COPY][PLUS

AN APPLE DISK UTILITY SYSTEM

COPY][PLUS COPYRIGHT 1982
Central Point Software, Inc.
P.O. Box 19730-#203 / Portland, OR 97219
PHONE 503/244-5782
THE SOURCE ST2008

APPLE is a trademark of Apple Computer, Inc.

TABLE OF CONTENTS

1. Introduction

Starting Up........................Page 1-1

2. DOS Utilities

New Disk Info......................Page 1-3
Catalog............................Page 1-4
Copy...............................Page 1-6
    Copy Files.....................Page 1-7
    Copy Disk......................Page 1-10
    Copy Dos.......................Page 1-11
Delete.............................Page 1-12
    Delete Files...................Page 1-12
    Delete Disk....................Page 1-13
    Delete DOS.....................Page 1-13
Lock/Unlock Files..................Page 1-13
Rename Files.......................Page 1-14
Format A Disk......................Page 1-15
Verify.............................Page 1-15
    Verify Disk....................Page 1-16
    Verify Files...................Page 1-16
    Verify Identical Files.........Page 1-17
    Verify Drive Speed.............Page 1-18
Track/Sector Map...................Page 1-19
View Files.........................Page 1-22
Fix File Sizes.....................Page 1-24
Change Boot Program................Page 1-24
Undelete Files.....................Page 1-25
Sector Editor......................Page 1-26
Boot Disk..........................Page 1-28

3. Bit Copy Program

Introduction.......................Page 2-1
The Basics.........................Page 2-1
Apple Disk Hardware................Page 3-1
Protection Schemes.................Page 4-1
Nibble Editor......................Page 5-1
Parameters.........................Page 6-1

4. Appendix A: Bit Copy Parameters...Page A-1
5. Appendix B: Decimal to Hex.......Page B-1

Introduction

With Version 4.0, Copy][Plus includes both a powerful DOS disk utility package and a sophisticated Bit Copy program. The DOS utilities allow you to manipulate DOS files and disks quickly and easily. The Bit Copy program can make backups of valuable software that has been copy-protected. In this manual, we´ll show you how to use each option step-by-step. In nearly every case, Copy][Plus will show "reminder" prompts as to what commands or menu options are valid. We encourage you to carefully read through this manual to take advantage of all of Copy][Plus´s features. We assume you are generally familiar with DOS and the standard DOS commands. Apple´s Disk Operating System manual and the book "Beneath Apple DOS" by Quality Software are both excellent sources of information.

If you are primarily interested in the Bit Copy program, you may wish to skip over to Chapter Two, which begins the discussion of making backups of copy protected software.

Starting Up

To access either the DOS utilities or the Bit Copy program, begin by booting the Copy][Plus disk. In a couple of seconds, the Copy][Plus System Menu will appear. Along the left side of the screen are the 16 main options. With some of these options are sub-menus to select specific functions.

Throughout Copy][Plus, you can press the Escape key ([ESC]) to back safely out of the sub-menu or current option.

To select an option, simply press the left or right arrow keys until the desired option is displayed in inverse, then press the return key. Note the prompt in the upper-right:

```
USE ARROW KEYS &
[CR] TO SELECT OPTION
```

([CR] stands for Carriage Return, and refers to the return key.)

Status Display

At the bottom of the screen will be the following display. (The slot numbers shown will be the slot used to boot Copy][Plus.):

```
DISK  SLOT  DRIVE  DOS  FREE&USED    PRINTER
 A     6      1    3.3                 OFF
 B     6      2    3.3
```

Copy][Plus uses names for the disk drives being accessed. The names are simply "A" and "B". In the above example, drive A refers to the disk drive in slot 6, drive 1. Drive B is slot 6, drive 2. Rather than asking for slot and drive numbers for every option, Copy][Plus allows you to quickly select drive A or B as a menu option. Any time you need to (for example, if you have more or less than two disk drives), you can change drives A and B to be different drives using the "New Disk Info" option.

You can also access disks in either DOS 3.3 or DOS 3.2 format. In the example, both drives are selected to read DOS 3.3 disks. If you want to read a DOS 3.2 disk, simply choose the drive you will use and select it as DOS 3.2 with "New Disk Info".

"FREE&USED" displays how many free and used sectors are on disks A and B. If a disk has not been read, this will be blank.

Bit Copy

To load the Copy][Plus bit copy program (for copying protected disks), press the arrow keys until the words "BIT COPY" are displayed in inverse, then press CR. Insert the Copy][Plus disk in the drive it was originally booted in, then press CR again. The bit copier will be loaded into memory and run.

New Disk Info

To select the New Disk Info option (second from the bottom), press the right or left arrow keys to display this option in inverse, then press CR. The slot number for drive A will begin flashing. If you want to change the slot number, you can type a new number, and it will replace the old. If you want to keep the current slot number, just press CR. (If you enter an invalid slot number, the Apple speaker will beep.)

Next, the drive number for drive A will flash. You can enter a new drive number or press CR to accept the one displayed.

The second digit of the DOS version will flash next. This allows you to select between DOS 3.3 or DOS 3.2. You can enter a 3 to select 3.3, a 2 to select 3.2, or press CR. If, for example, you select drive A as DOS 3.2, then you will want to

insert your 3.2 disks in drive A to be read. DOS 3.2 and 3.3 are not compatible with each other. This means that if a DOS 3.2 disk is in a drive selected as DOS 3.3 (or vice versa), any disk access will cause I/O errors.

After selecting the slot, drive, and DOS numbers for drive A, repeat the process for drive B. If you have only one disk drive, you will want to select drive B as the same disk drive as A.

Next, the word "OFF" underneath the PRINTER label will change to "SLOT 0", and the "0" will flash. Copy][Plus allows you to send catalog listings and file data displays to your printer if desired. If you intend to use the printer, type the slot number that the printer interface card is in. If you don´t want to use the printer, press CR or type "0". The zero is used to designate "no printer", since printer cards cannot be used in slot 0.

After selecting the printer slot, the main menu will become active again. (If a printer was not selected, the slot number will change back to "OFF".) You´re now ready to use the drives you´ve designated as A and B.

Catalog

To display or print a catalog of the disk, select the CATALOG option. A sub-menu will appear on the right of the screen. The options are:

Normal
With file lengths
With deleted files
With hidden characters

Once again, use the arrow keys and CR to choose the option. Next you will be asked if the catalog is for disk A or B. Select the drive with the arrow keys and CR. If the printer is "off", the catalog will be displayed. If the printer is "on", you will be asked whether or not you want a printout of the catalog. Answer Y (yes) or N (no).

Normal

The "normal" catalog is similar to the standard DOS "CATALOG" command. The disk volume number is shown, then for each file, the optional "locked" asterisk, the filetype letter, the file length (in sectors), and finally the filename are shown, one line for each file. The catalog pauses after every 20 files. You can continue by pressing any key (except [ESC], which will stop the catalog and return you to the main menu). If the catalog is being sent to the printer, it will not pause.

Note: The Copy][Plus disk is not itself a standard DOS disk. If you try to get a catalog of the disk, or do any DOS access to the disk, you will get an I/O error.

With File Lengths

The catalog "with file lengths" shows all the same information as the normal catalog. For all Basic files, it also shows the actual length of the program in bytes, using both decimal and hexadecimal notation. For binary files, it shows both the starting memory address of the file and its length. Here is an example catalog listing for a couple of files:

```
*A 006 HELLO
       L1137  (L$0471)
*B 003 CHAIN
       A2056, L456  (A$0808, L$01C8)
```

This shows that the Basic file HELLO is 1137 bytes long ($471 in hex), and the binary file CHAIN has a starting address of 2056 and a length of 456 (with corresponding hex numbers in parentheses.)

With Deleted Files

The catalog "with deleted files" includes the files on the disk which have been marked as deleted, but have not yet been overwritten by a new file entry. Any deleted files are marked in this display with the letter "D" to the left of the entry. (Note that in some cases, deleted files can safely be recovered and made active again using the UNDELETE FILES option, explained below.)

With Hidden Characters

A catalog "with hidden characters" allows you to see any imbedded control characters, which are normally not printed by Copy][Plus. The control characters show up as flashing characters. Lower-case characters are also changed to upper-case, so they can be read on the Apple's upper-case only display. If the printer is on, lower-case characters are instead unchanged, and any control characters are also translated to lower-case.

Copy

The main COPY option gives you three separate choices:

```
Copy Files
Copy Disk
Copy DOS
```

For each of these choices, you can copy from disk A to disk B, or from B to A, selected by the next menu:

```
SELECT DISK:

A TO B
B TO A
```

If you have only one disk drive and have used New Disk Info to set both A and B to that drive, then it doesn't matter whether you copy "A TO B" or "B TO A".

Copy Files

The copy files option allows you to copy standard DOS files from one disk to another quickly and easily. You should have the disks in the drives before finishing the menu selections. (If you have only one drive, you should have the source disk, the one containing the files to be copied, in the drive.) The source drive will whir for a moment, then a "catalog display" for the source disk will appear with a prompt below.

The catalog display is used in various ways throughout Copy][Plus for selecting files to be worked with. Here it is used to determine which files to copy. Note that the first file in the catalog is displayed in inverse. By using the arrow keys, you can cause any file in the catalog to be in inverse. This inverse file is considered the "current" file. If you repeatedly press the

arrow keys, the display will scroll.

The prompt below the display reads:

```
[CR]-MARK, [D]ELETE, [E]NTER FILENAME,
NUMBER-INSERT, [G]O, [ESC]-EXIT
```

These commands allow you to select not only which files to copy, but in what order to copy them. This is a handy feature if you want files to appear in a certain order on the catalog of a disk.

Pressing CR will place a number to the left of the current (inverse) file. The first CR will place the number 1, the second a 2, etc. These numbers represent the order the files will be copied in. If you accidentally press CR to number a file you don't want to copy, you can remove the number by moving the inverse field to that file and pressing "D" for Delete Number. You can also make insertions in the list of numbers by typing a number directly, then pressing CR.

In addition, you can select one or more files by pressing "E", for Enter filename. You can type a single filename, and the program will look for that name in the catalog display and mark it with the next available number. You can also enter filename "patterns".

A pattern is a filename with one or more equal signs ("=") in it. The equal sign is a special "wildcard" character that will match any number of characters in the catalog, as long as the rest of the filename matches. For example, the pattern "AB=" will match the files "AB", "ABCDE", and "ABRAHAM". The pattern "=N=" will match the files "N", "OH NO", or any filename containing the letter N. The pattern "=" will match anything, and can be used when you want to copy every file on the disk.

In addition, patterns can specify what filetypes to match. If you want a pattern to match only certain filetypes, finish the pattern by typing a comma, followed by the filetype letters used in the catalog:

A - Applesoft
I - Integer
B - Binary
T - Text

For example, the pattern "=XYZ,BT" will match any file whose name ends in "XYZ" and is a binary or text file. The pattern "=,A" will match any Applesoft file.

After entering the pattern and pressing CR, the program will scan through the display, marking all matching files. The inverse field will then jump to the last file matched. If no files match, the inverse field will return to the file that was in inverse before you pressed "E".

When you've selected all the files that you want to copy, press [G] for Go to begin the copy. If only one drive is being used, you will be prompted to insert the proper disk.

The first file to be copied (or as much of it as will fit in memory) will be read, then the catalog on the destination disk will be read. At this point, the program will check to see if any of the files being copied already reside on the destination disk. If not, copying will continue. If there are duplicate filenames, you will be prompted, as in this example:

```
FILE HELLO
ALREADY EXISTS. NOW WHAT?

[C]OPY ANYWAY, [N]EW NAME, [D]ON'T COPY,
[ESC]-EXIT COPY
```

(If the duplicate file is locked, the program will say "IS LOCKED" instead of "ALREADY EXISTS".)

If you select to Copy anyway, the original will be deleted, then the new file copied. If you select New name, you will be asked to type in a new name for the file. The Don't copy option will simply not copy this file, and pressing ESC will abort the entire copy option.

Note that if more than one duplicate file is on the destination disk, Copy][Plus will ask about all of these files before doing any copying. This means that once the questions have been answered, the program will copy all of the files without requiring your attention.

As the files are being copied, they are displayed in the "file queue", a straightforward list of the files, with the file currently being copied displayed in inverse.

Copy Disk

Copy Disk is a fast routine for copying any standard 16 or 13 sector diskette. Copy Disk automatically formats as it copies, so disks do not have to be formatted ahead of time. To copy a disk, simply select the option, insert the disks, and press CR. Note that the DOS version must be the same for both drives A and B. If for some reason you wish to stop the copying, pressing ESC will return you to the menu.

As the Copy Disk option makes the copy, it checks for errors. If an error occurs, a message will be displayed showing what kind of error it is (Read error or Write error) and what track on the disk it occurred on. The program will continue copying the rest of the disk. A read error means that one or more sectors on the source disk are unreadable. The disk media itself may or may not be damaged. If a write error occurs, then the media on the destination disk is most likely damaged. Double-check everything, then try again.

Even if the Copy Disk routine reads a bad sector, it might still write a "good" sector to the destination disk. That is, the data in that sector may be inaccurate, but an I/O error will often not occur if that sector on the destination disk is read.

If a disk is getting old and begins to create I/O errors, the data should be copied to a new disk using Copy Disk.

Copy DOS

Copy DOS is similar to Copy Disk, but it only copies the first three tracks of a disk. This is where the Disk Operating System is written on DOS disks. Copy DOS can be used to make "master disks" which will boot on any Apple, regardless of how much memory is available.

To turn a "slave" disk into a "master" disk, insert a DOS System Master disk in the source drive and the disk to be "mastered" in the destination drive, then select the Copy DOS option. The master DOS will be copied onto your disk. The master DOS will try to run an Applesoft

program with the name HELLO when the disk is booted. If you want the disk to run a different program when booted, use the Change Booting Program option, discussed below.

Delete

The main DELETE option has three sub-options:

Delete Files
Delete Disk
Delete DOS

Delete Files

This option is equivalent to the standard DOS "DELETE" command, except that a number of files can be deleted at one time. After selecting the Delete Files option and drive A or B, a catalog display appears, similar to the one used in Copy Files. The prompt reads:

```
[CR] TOGGLES MARKER, [E]NTER FILENAME,
[G]O, [ESC]-EXIT
```

Pressing CR causes an arrow "->" to appear to the left of the file entry. The arrow marks the file to be deleted. Repeatedly pressing CR toggles the arrow on and off. A number of files are marked by using the arrow keys and CR. A filename or pattern can also be entered with [E]. The rules for the pattern are the same as for Copy Files. Any file that matches the pattern will be marked to be deleted.

To carry out the deletion, press [G] for Go. All files marked will be deleted. The "file queue" display will show the filenames as the files are deleted.

Delete Disk

The Delete Disk option cleanly erases all the "record-keeping" information on the disk, including the names and locations of the files, and the presence or absence of DOS. Deleting a disk is equivalent to reformatting it to start over, but takes less time. (An unformatted disk, however, must be formatted before it can be used.)

An extra warning prompt will appear on the screen to prevent data from inadvertently being destroyed:

```
DELETE DISK (Y/N) ?
```

Answer "Y" to delete the disk.

Delete DOS

As mentioned above, DOS uses the first three tracks on a disk. The Delete DOS option "frees" two of those tracks so that files can use them. The first track (track 0) is not accessible to files, and is not freed. Deleting the DOS increases the storage capacity of a disk by 8 Kilobytes, but the disk cannot be booted, since there is no longer any DOS to boot. (If you try to boot a disk after running either Delete DOS or Delete Disk, you will see a message saying that this disk does not contain DOS and is unbootable.)

Lock/Unlock Files

If you wish to lock or unlock one or more files, choose this option and select disk A or B. The drive will whir and a catalog display for the disk will appear. As in a normal catalog, an asterisk to the left of the filetype letter designates each file that is locked. A new prompt is displayed:

```
[CR]-TOGGLE ASTERISK, [E]NTER FILENAME,
[G]O, [ESC]-EXIT
```

Use the arrow keys to select a file, then press CR to toggle its ´locked´ asterisk on or off. You can use these keys to set the desired locked status for every file on the disk.

To lock or unlock a number of files automatically, press [E]. You´ll be prompted for a filename, with the same pattern capabilities as discussed above. After entering a filename, you will see:

```
[L]OCK OR [U]NLOCK?
```

Press "L" to lock all of the files that match the pattern; press "U" to unlock them.

After setting all of the desired files, press "G" for Go. The catalog will be written back to the disk, with the proper files locked and unlocked.

Rename Files

To rename files, select this option and select disk A or B. The usual catalog display will appear, with another prompt:

```
[CR]-SELECT TO RENAME, [E]NTER FILENAME,
[G]O, [ESC]-EXIT
(RENAMED FILES ARE MARKED)
```

To rename a file, move the inverse field to that file with the arrow keys, then press CR. You will be asked what to rename the file. Enter a new name and press CR. This must be a legal DOS filename; i.e. it must begin with a letter and cannot contain a comma. If you enter a bad filename, the warning message "INVALID FILENAME" will appear and you will be prompted for another filename. If you decide that you do not want to rename the file, press ESC.

For every file that is renamed, an arrow ("->") appears to the left of the file. This simply serves as a reminder as to which files have been renamed.

The Enter filename option is available, but since files must be renamed manually, the "E" option stops at the first file that matches the pattern, leaving that file displayed in inverse. From here you can press CR to rename the file.

To make the changes permanent, press "G". The new filenames will be written to the disk.

Format Disk

This option formats a disk so that data can be written to it. It is similar to the DOS command "INIT", except that DOS is not written onto the disk and a Basic program is not saved. (These can be duplicated with the Copy DOS and Copy Files options. A formatted disk will not boot unless DOS and a greeting program are copied onto the disk.)

After selecting the disk to be formatted, an extra prompt message will appear, to verify that you want to format the disk. The disk will format using whichever DOS is selected (from New Disk Info) for that drive.

Verify

The Verify option is used to select one of four sub-options:

1. Verify disk
2. Verify files
3. Verify identical files
4. Verify drive speed

Verify Disk

This option is used to check if any sectors on the disk are bad. It reads each of the 35 tracks (numbered 0 to 34) in turn. If any bad sectors are found, their track and sector numbers will be displayed. When finished, the program will show the total number of errors. If you want to exit the verify option, you can press ESC at any time.

Verify Files

Verify Files checks the data and sectors used by individual files. After selecting the disk, the drive will whir and a catalog display will appear. Here, the files to be verified can be selected with CR the same way the files to be deleted were selected in the Delete Files option. An arrow will appear by all selected files. The Enter filename command can also be used to select files, with the usual multi-file pattern capabilities. To begin verifying those files, press "G".

The file queue display will show each file in inverse as it is verified. If an error occurs, the track and sector number for the error will appear. You can press CR to continue verifying the file, the space bar to move to the next file, or ESC to return to the main menu.

Apple disks have 35 tracks, each consisting of 16 sectors for DOS 3.3, or 13 sectors for DOS 3.2. If an error occurs showing a track number of 35 or more, or a sector number of 16 or more (13 for DOS 3.2), this indicates a data error in either the catalog itself or the file's "Track/Sector List", since such a track or sector can't exist on the disk. (See Appendix C of the Apple DOS Manual for more information.)

Verify Identical Files

This option determines whether or not two files on different disks are identical. This is useful when you have files on different disks with a similar name, and you don't know whether they are copies of the same file or are different.

To use Verify Files, insert the disks in A and B, and select the Verify Files option. Single drive users should insert one of the two disks in the drive. You will be prompted when to switch disks.

A catalog display for disk A will appear. To select the desired file from this disk, use the arrow keys to move the inverse field to the file, then press "G". The file will be read into memory, then a catalog display for disk B will come up. In the same way as for the first disk, select the file to be checked. If the files being compared are text files, you will then be asked:

```
[S]EQUENTIAL OR [R]ANDOM ACCESS
TEXT FILES?
```

If you select Sequential text files, the comparison will stop if it reaches an end of file marker, so data beyond the end of the file is not compared. If you select Random Access text files, the end of file markers will be ignored and every data sector will be checked.

After the files are compared, a message will appear informing you of whether the files are identical or different. If they are different, the message will say how many bytes into the file the first difference was found. (If you want, you can then use the View Files option, discussed below, to see what the difference is.)

If the files have different filetypes (e.g. if one file is a Basic program and the other is a textfile) they cannot be compared, and the following message will be displayed:

```
THE FILES HAVE DIFFERENT FILETYPES
```

Verify Drive Speed

To properly read the data on disks, the disk drive must spin at the right speed. This speed is 5 revolutions per second, or 1 revolution every 200 milliseconds. This speed was set at the factory, but with time, the drive speed can drift. If the speed is too far from 200 milliseconds, I/O errors can occur, or data can be written that is unreadable on a normal-speed drive.

The Verify Drive Speed option allows you to periodically check the speed of your disk drives. Select the option and disk A or B, then insert the Copy][Plus disk (or any blank disk) in the appropriate drive and press CR. (Do not use a valuable disk. This option will erase track 34!) In a few seconds, the drive speed will be displayed. Note that the proper drive speed can vary from 198 to 202 milliseconds (ms.). Small fluctuations in the speed are also normal. The speed will be displayed until you press ESC.

If the speed is out of the above range, this are stored. There should also be a stripe along the left side. If the disk contains DOS, then the stripe will cover tracks 0, 1, and 2. If not, the stripe will cover only track 0. (Track 0 is unavailable for file data storage.)

Possible problems:

If there are still inverse spaces in other portions of the grid, this represents other sectors on the disk which are marked as "in use", but are not being used by any file. If the disk is a commercial product, it is possible that the sectors are being used for storing some kind of special data. They are marked as in use to keep them from being overwritten. This is rare.

If a file uses a sector that is not marked "in use", a plus sign appears in that sector of the grid and the warning message "FREE SECTOR" and the filename are printed. This means that the sector could be overwritten if more data is written to the disk. The file should be copied to another disk. You have the option to continue the Track/Sector Map or return to the main menu.

If two files reference the same sector, a plus sign will appear in that sector, and the message "SECTOR CONFLICT" will appear. Since two files cannot (legally) use the same sector, at least one of the files is damaged. You should copy both files to a new disk and check them for accuracy.

If the error "INVALID NUMBER" occurs, with a track or sector number out of range, this means that either the catalog information or a track/sector list is damaged. If no letters for that file appear yet on the grid, then it is the catalog information that is in trouble, and the file is lost. Otherwise, the track/sector list is bad,

and some of the file may still be readable.

The errors discussed above do not occur very often. The Track/Sector Map can be used to help understand disk usage, and catch possible errors before they crash a disk or destroy additional data.

View Files

The View Files option allows you to quickly and easily look at the data in any file. This is useful for double-checking exactly what is in a file before copying it, deleting it, etc. View Files has two sub-options, for viewing the data as values or as text. The values option shows both the hexadecimal numbers and the ASCII characters in the file. The text option prints just the characters in a more readable form. In addition, if the printer is selected, the data can be sent to the printer.

To view one or more files, select the View Files option, the Values or Text sub-option, and disk A or B. A catalog display for the disk will appear. Use the arrow keys and CR to select the files you want to view, then press "G". If the printer is "on", you´ll then be asked whether or not you want a print-out of the files. Answer "Y" for Yes to get a print-out.

For each file in turn, the filename will be displayed. The file is displayed a page at a time. You can press CR to view the file, ESC to return to the main menu. If you´ve selected to view two or more files, you can also press the space bar to advance to the next file.

When using the View Values option, the file is displayed as hexadecimal bytes, 8 bytes per line, with the equivalent ASCII characters to the right. Control characters are replaced with periods. In the View Text mode, the characters are printed out in standard 40-character lines. Control characters are not printed, except for carriage returns.

In the upper right portion of the screen is a running "byte count", showing how many bytes in the file have been printed. This can be used to find the approximate locations of strings or bytes in the file. Remember that the Verify Identical Files option can be used to find at which byte two files differ. This difference can then be found by keeping an eye on the "byte count" while viewing the files.

At the end of the file, there may be a few funny characters, including inverse "@" signs. These are extra characters beyond the end of the real end-of-file. They were not suppressed because random access text files have end-of-file markers interspersed throughout the file, before the file has actually ended. These files can still be viewed. The View Files option stops reading when there are no more data sectors to be read.

Fix File Sizes

When a short program is saved over the top of a long one, Apple DOS does not free the extra sectors that are no longer used. They continue to use space on the disk. Usually, the only way to recover the space is to load a file, delete it, and save it back to disk.

The Fix File Sizes option is designed to recover unused space at the end of files. It will free extra space from Basic and Binary files, but not textfiles. As mentioned above, the true lengths of textfiles cannot readily be determined, because random access textfiles can have end-of-file markers anywhere in the file.

To free up extra space, simply select Fix File Sizes and disk A or B. The "file queue" will show each file in turn as the program checks the file and recovers any extra space. The file queue will quickly skip over any textfiles on the disk, as they are not checked.

Change Boot Program

When a standard initialized DOS disk is booted, it automatically runs whatever Basic program the disk was initialized with. For example, a disk that was initialized with the command "INIT HELLO" will run the program "HELLO" whenever it is booted. Using the Change Booting Program option, you change the DOS to boot a different Basic program, or even BRUN a binary file or EXEC a textfile on boot-up.

Select the Change Booting Program option and disk A or B. A catalog display for the disk will appear. At the bottom of the screen, the name of the file that the disk currently boots up with will be printed. To select a new booting program, use the arrow keys to place the inverse field over the desired file. You can also Enter a filename or a pattern. The inverse field will stop at the first filename that matches the pattern.

Press "G" to save this file as the booting program. Copy][Plus will automatically check the filetype of the file, and set either the RUN, BRUN, or EXEC command for boot-up.

Undelete Files

When a file is deleted, it is not immediately erased. It is instead marked internally as a deleted file, and its sectors are marked as free for being re-used. If other data does not later overwrite part of the file, it can still be recovered and made an active file. If a file has just been accidentally deleted, and no other disk writing has occurred, the file can always be recovered, or "undeleted". That is what the Undelete Files option is for.

To undelete one or more files, select the option and disk A or B. A catalog display will come on the screen, this time containing a list of all the deleted files still stored invisibly in the catalog. (If there are no deleted files in the catalog, the message "NO FILES" will appear.) Use the arrow keys, the Enter filename command, and CR to select the files to be undeleted. Press "G".

The file queue will show the files as the program attempts to undelete them. If a deleted file has already been partly or completely overwritten with other data, Copy][Plus will not undelete it,

since the data is not recoverable. If any of the files cannot be undeleted, they will then be listed with the label "LOST FILES". You may now gasp or groan appropriately. The rest of the files, however, will be active.

Sector Editor

The Sector Editor allows you to directly view and modify the data on any sector of the disk. This is handy for people interested in poking around Track/Sector Lists, etc. to learn more or to fix problems. It is also used with the Bit Copy program for copying certain protected disks. You should use care when working with the Sector Editor, to avoid accidentally erasing or modifying important data on the disk.

To use the Sector Editor, select the option and disk A or B. The Sector Editor display will appear, with the sector buffer (256 bytes) cleared to zeros. Fifteen lines of the buffer are displayed at a time, consisting of a hex address on the left, 8 hex values, and the same values as ASCII characters on the right. The "double" cursor appears in inverse over both the first value and the first character.

The cursor can be moved through the buffer with the I, J, K, and M keys. "I" moves the cursor up, "J" to the left, "K" to the right, and "M" down. The buffer display will scroll up or down to keep the cursor on the screen. "B" moves the cursor directly to the beginning of the buffer; "E" moves the cursor to the end.

To read a sector from the disk, press "R" for Read. You will be prompted for the track and sector numbers of the sector to be read. These numbers should be entered in hexadecimal. An invalid character or an invalid number will cause the speaker to beep. After entering the track and sector numbers, the sector will be read into the buffer.

To write a sector back to the disk, press "W" for Write. You will again be prompted for track and sector. After entering the values, the sector will be written.

After the first sector is accessed, Copy][Plus remembers the track and sector used. When reading or writing, you can press CR without any numbers to accept the current values. You can also change values when writing, for moving data from one sector to another.

The data can be modified by entering new hex values or new text characters. To enter hex values, move the cursor to the appropriate place and press "H" for Hex. The cursor over the hexadecimal value will flash. Now enter the new value over the old. Pressing the space bar will advance you to the next byte, and pressing CR will take you out of the hex entry mode.

To enter characters, position the cursor and press "T" for Text. The cursor over the text character will flash. Type new characters to enter those characters into the buffer and advance the cursor. Press CR to finish text entry. Note: While in text entry mode, any control characters typed, including the arrow keys (but not including CR or ESC), will be placed directly into the buffer.

When copying protected disks, sometimes the Sector Editor is used to modify data on the copied disk. In a few cases, the data cannot be read without first making changes in the Copy][Plus disk read

routines. These changes can be made in the sector editor by pressing "P", for Patch Disk Read/Write. After pressing "P", the message "(Disk READ/WRITE IS PATCHED)" will appear at the bottom of the screen. Many protected disks can now be read. Pressing "P" again will return the disk Read/Write routines to normal, as will leaving the Sector Editor.

The separate pages that accompany this manual on copying specific disks discuss which disks need the special patch. The patch option should not be used unless specifically called for in the instructions for copying a disk.

Press ESC to exit the Sector Editor.

Boot Disk

When you want to exit Copy][Plus and run another program without turning your Apple off, select the Boot Disk option. You will be prompted to insert the disk to boot, using the same drive that Copy][Plus was booted in. Press CR and the disk will be booted.

This concludes the discussion of the DOS Utilities portion of Copy][Plus. The remaining chapters are dedicated to the Bit Copy program, which can be used to make backups of many protected diskettes.

Bit Copy Program

The Copy][Plus Bit Copy program is designed to allow you to make backups of software which, due to copy protection schemes, does not copy using standard disk duplication programs. The Bit Copy program is easy to use, yet is capable of being adjusted to handle nearly every type of protection scheme currently in use. It is modified through a comprehensive set of parameters which will be described in later chapters. Also enclosed with this manual is the current list of instructions for making backup copies of many popular diskettes.

The complete Copy][Plus package is much too large to fit into your computer's memory at once and still allow for any buffer space, so it has been divided into two parts: the disk utilities, and the Bit Copy program. The Copy][Plus disk, when booted, will display the menu for the complete package, which should look similar to:

```
                  COPY ][ PLUS 4.0
     (C) 1982 CENTRAL POINT SOFTWARE, INC.
 -----------------------------------------
 CATALOG DISK                USE ARROW KEYS
 COPY                        & [CR] TO SELECT
 BIT COPY                    FUNCTION
 DELETE
 LOCK/UNLOCK FILES
 FORMAT DISK
 VERIFY
 TRACK/SECTOR MAP
 VIEW FILES
 FIX FILE SIZES
 CHANGE BOOT PROGRAM
 UNDELETE FILES
 SECTOR EDITOR
 NEW DISK INFO
 BOOT DISK

 DISK  SLOT  DRIVE  DOS   FREE & USED  PRINTER
  A     6      1    3.3                  OFF
  B     6      2    3.3
```

Selecting the Bit Copy option from this menu will load Copy][Plus from the disk. The disk has been specially formatted for fast booting, so you will not spend time waiting for Copy][Plus to load. After selecting the Bit Copy option, you should see this sign-on screen:

```
        COPY ][ PLUS BIT COPY PROGRAM
    (C) 1982 CENTRAL POINT SOFTWARE, INC.
 -------------------------------------------
    COPY OR EDIT (C OR E)  DEFAULT=COPY
```

The above prompt is asking whether you wish to scan (edit) or backup (copy) a disk. Copy][Plus contains a nibble editor which allows you to scan a disk to see what types of protection it uses, and to change parameters to adjust Copy][Plus to copy the disk. The Nibble Editor's operation is described in a later section. Before we go any further, let's make a backup of your Copy][Plus diskette, so that if disaster strikes you will be ready. (Note that Central Point Software will re-make your original Copy][Plus diskette if it ever fails to boot for any reason. If the diskette has been physically damaged, you will be charged the cost of a replacement disk.)

Since we are going to copy a disk, just press the return key to take the default. Next, you will be asked the following questions:

```
  ORIGINAL SLOT:   6
         DRIVE:   1

  DUPLICATE SLOT:  6
         DRIVE:   2
```

These specify which drive to copy from and to. The slot number is the slot in your Apple which contains the disk controller card. Each controller card can handle two drives. Usually, only one controller card is used, so the slot options usually won't change. The default value (the value displayed in inverse video on your screen) is not fixed, and is the slot of your boot drive. Therefore, if your controller card is plugged into slot '4', this will automatically be displayed as the default value, rather than '6'. If you have only one drive, you will want to let Copy][Plus know by specifying a '1' for the DUPLICATE DRIVE.

So, if you have a single disk drive, and your controller card is plugged into slot six, after you have answered the above questions, your screen

should look like this:

```
  ORIGINAL SLOT:   6
         DRIVE:   1

  DUPLICATE SLOT:  6
         DRIVE:   1
```

When using Copy][Plus with a single disk drive, it will instruct you to swap the original and duplicate disks at the proper time.

Next, you will be asked which track to start copying on. Ordinarily, this will be '0'. Sometimes, however, a disk may not use every track, and you may wish to bypass the unused tracks. All disks will use track '0', however, and the Copy][Plus disk is no exception, so take the default and press the return key. Your screen should show:

```
  ENTER START TRACK:  0
```

Now, Copy][Plus wants to know how far into the disk you wish to copy. Many disks (especially games) don't need the entire capacity of the disk and will only use the first few tracks. Disks for the Apple usually have 35 tracks, numbered in hexadecimal from 0 to 22. (See the hexadecimal to decimal conversion chart at the end of this manual if you are not familiar with this numbering system.) The Copy][Plus disk only uses the first 16 tracks, which is track 'F' in hexadecimal. Enter this number here. You should now see:

```
  ENTER END TRACK:  F
```

Lastly, you need to let Copy][Plus know what the spacing is between each track on the disk.

Normally, a spacing of ´1´ is used, but it is possible to enter any number here between ´.5´ and ´21´. The tracks on the Copy][Plus diskette utilize standard track spacing, so take the default and press ´RETURN´. The display should look like this:

```
  ENTER TRACK INCREMENT:  1
```

At this point, we have told Copy][Plus all it needs to know about the disk drives and also what parts of the diskette contain information that needs to be copied. Since the Copy][Plus diskette does not use standard Apple formatting, we need to describe the formatting it does use. This is done through Copy][Plus ´PARAMETERS´. Looking at the bottom of the screen, you should now see this message:

```
           -- INSERT DISKETTES -

RETURN TO BEGIN             Q TO QUIT
ESC TO RESTART              / TO MODIFY
```

The ´/´ Command tells Copy][Plus we wish to change some of the default parameters. After you press ´/´, you will see:

```
           -- PARAMETER CHANGE --

       CHANGE WHAT PARAMETER
```

Each Copy][Plus parameter has a number. These numbers start at zero and extend upwards, and, like the track start and end, are numbered in hexadecimal. After each parameter number entered, you will be asked to enter its new value. The current value will be displayed underneath the flashing cursor on the screen. If you press ´RETURN´, the current value will not be changed.

```
           -- PARAMETER CHANGE --

         CHANGE WHAT PARAMETER:
                 TO WHAT VALUE:
```

To backup Copy][Plus, we need to change 4 parameters. Here are their numbers and new values:

PARAMETER #	DEFAULT	NEW VALUE
E	D5	94
F	AA	92
10	B5	CD
A	9	3

Enter each parameter number and press ´RETURN´. When you see the ´TO WHAT VALUE´ prompt, type in the ´NEW VALUE´ number from the above table. Remember, numbers in Copy][Plus are entered in hexadecimal, so they may not always look like numbers.

Once you have finished changing these parameters, Copy][Plus is ready to make your backup. Press ´RETURN´ once more and you will be taken back to this menu:

```
           -- INSERT DISKETTES --

RETURN TO BEGIN             Q TO QUIT
ESC TO RESTART              / TO MODIFY
```

Insert a blank diskette (it doesn´t need to be formatted first) into the disk drive you specified as the ´duplicate drive´, and make sure your Copy][Plus disk is still in the ´original drive´. Press ´RETURN´ and your backup will be made.

Copy][Plus uses the middle of the screen to give you detailed information about each track as it is read and analyzed, and the bottom of the screen to give you status information. Copy][Plus goes through 4 main stages when copying each track. First it must ´read´ each track into memory from the original disk, then it must ´analyze´ the track before ´writing´ it out to the duplicate drive. Lastly, it must ´verify´ that the track was written correctly, then it can go on to the next track. As the copy process continues, you will see the following characters appear on the track/status display on the lower portion of the screen.

R	READING TRACK
A	ANALYZING TRACK
W	WRITING TRACK
V	VERIFYING TRACK

In addition, as each track is finished, a track status (error) number will be left on the display. The codes, and their meanings are:

0	NO ERROR. TRACK COPIED CORRECTLY.
1	READ VERIFY WARNING. TRACK MAY NOT BE VALID.
2	READ ERROR. PARAMETERS INCORRECTLY SET.
3	TRACK LENGTH GREATER THAN TRACK MAX.
4	DUPLICATE DISK WRITE PROTECTED. REMOVE TAB.
5	WRITE VERIFY ERROR.
6	NIBBLE COUNT ERROR (ONLY IF SELECTED).

You should see a series of zeros forming along the bottom of the screen, indicating a correct copy. If you do not, you may have entered the parameters incorrectly, or you may need to adjust your drive speed.

You now know the basics of the Copy][Plus Bit Copy program. Many diskettes can be backed-up using just the defaults, with no parameter changes. More recently, however, software companies have found new and clever ways of protecting their software, so it is quite possible that many of your protected diskettes will require parameter changes. With this manual is supplied a list of parameter changes for many popular programs. Using the information in this section, and this list, you should be able to backup many of these diskettes. If a program you wish to backup is not on the list, always try the default settings first. If that doesn´t work, you may wish to use the Copy][Plus Nibble Editor to analyze the diskette. (See the Nibble Editor section (chapter 4) for more information.)

The next chapter is dedicated to the Apple disk hardware and recording techniques. Chapter 5 introduces protection schemes and Chapter 6 includes a detailed description of the Copy][Plus parameters, how they work, and the internal routines they control. These parameters are also listed numerically in Appendix A. If you don´t mind getting your feet a little wet, dive in! We´re sure you´ll find the information helpful.

Disk Hardware

This chapter will attempt to give some background on the Apple disk drive hardware and recording schemes that will be necessary for do-it-yourself parameter hunters. (Those adventuresome souls who figure out the parameter changes for specific diskettes.) We won´t cover all there is to know, but will hopefully give you enough information to use Copy][Plus to its maximum potential.

The Apple disk hardware is quite different from that of many other popular micro-computers. In fact, with the advent and widespread use of ´disk controller chips´, it may not be long before nobody really knows "what goes on in there". In the meantime, the Apple provides an excellent opportunity for software companies and anyone else to devise their own recording schemes, as the hardware is so versatile. This is precisely why protection schemes are often much better on the Apple Computer than on their counterparts. With the ability to do almost anything (within certain limitations), it is quite possible to create a disk that doesn´t behave much like an Apple disk at all.

If you already understand standard Apple disk formatting, and know what a ´sync byte´ is, then you may wish to skip this section. The next chapters describe the Copy][Plus parameters and algorithms and how all this relates to copy protection schemes.

Still with us? Good. Before we start looking at tracks and sectors, let´s take a look at the information encoding that occurs when writing to a disk drive, and the limitations that the drive hardware imposes.

When information is written to a diskette, it is written serially. What this means is that each ´bit´ of every ´byte´ to be written to a diskette is sent out one at a time, until the entire byte has been transmitted. To complicate matters, disk hardware is not capable of reading too many zero bits in a row, so the computer must first ´encode´ the information in a manner that makes it acceptable to the disk hardware.

There are several methods of encoding data for disk drives, and three are commonly used on Apple diskettes. The difference in storage capacity between DOS 3.2 and DOS 3.3 was a direct result of a small hardware change and the use of a new encoding scheme. While we won´t go into these schemes here as they don´t affect disk protection (if you´re interested, read Beneath Apple DOS by Quality Software), we should mention the third scheme as it is used in the address header of each sector. Being able to read this information is quite helpful.

As we mentioned above, disk hardware can´t reliably handle data with many consecutive zero bits. Just how many depends upon the particular hardware, but the following encoding scheme has been widely used is both highly reliable and wasteful of space. It is, however, simple and quickly decoded, which is why it is used in the address headers of sectors. Since there is not much information, any wasted space is minimal. This scheme is a standard two-for-one encoding method and works like this:

If I have one byte of information, and I want to write it without ever having two consecutive zeros, I can split it into two parts, putting all the odd bits into one byte and all the even bits into another, while setting each unused bit to a one. For example, a ´1´ as an eight bit binary number would look like:

```
0 0 0 0 0 0 0 1
```

which has seven consecutive zero bits - altogether unacceptable to our disk hardware. So, let´s take the number and shift it right once and set all the even bits to ´1´. This would look like:

```
1 0 1 0 1 0 1 0
```

which our picky disk hardware would love. Notice that only 4 bits contain real information. Let´s do the same again without a shift. This would look like:

```
1 0 1 0 1 0 1 1
```

Notice that our encoding scheme guarantees that there will never be two consecutive zeros because it splits up the byte into 2 bytes, setting half the bits to ones. Hold onto this knowledge as we´ll be using it in a moment when we discuss sector addresses. But for now, back to more familiar territory...

Let´s take a look at a normal DOS 3.3 diskette. We´ll start off about where the Apple disk manual leaves you, so if it gets confusing, you might want to refer to your Apple DOS manual, Appendix C.

As you know, an Apple disk drive running under DOS 3.3 can store approximately 140K (143,360 bytes) of information. This is organized as 35 tracks with 16 256-byte sectors per track. This means that each track can hold a little over four thousand bytes of your program or data (16 X 256 = 4096). How much of that is really available? It might not be as much as you think. DOS itself takes 3 tracks (tracks 0-2) and the catalog (disk directory) takes up another (track 17). Therefore 16K is really not free, even on a freshly initialized diskette, leaving you with about 124k (126,976 bytes) of storage space.

So each track on a diskette then is really made of 16 smaller "blocks" of information called sectors. How does the hardware know which one to get when you ask for a program? It does this by breaking each sector down into two parts: a data portion (the part you know about) and an address portion that, like any other address, identifies this sector from all the others.

An Apple sector actually begins with the address portion, and the data portion of the sector follows right behind. The address portion contains such useful information as the diskette volume number, the track that the sector is on, the sector number itself, and just to make sure everything is OK, a checksum of all the above information. By using a checksum, you can be reasonably sure that DOS won´t write to sector 7 when it should have written to sector 1.

Now that we have this basic information behind us, let´s take a look at a standard DOS 3.3 sector, and describe each part as we go along. Don´t lose heart, this table of numbers really does mean something!

```
FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF      <- SELF-SYNC FIELD
FF FF FF FF FF FF FF FF
D5 AA 96 FF FE AA AA AA      <- SECTOR ADDRESS
AA FF FE DE AA EB FF FF
FF FF FF FF FF FF FF FF      <- SELF-SYNC FIELD
D5 AA AD 96 96 96 96 96
96 96 96 96 96 96 96 96
. . .                        <- SECTOR DATA
96 96 96 96 96 96 96 96
DE AA EB FF FF FF FF FF      <- END OF SECTOR
```

We've put some labels out to the right to help us along. Let's start by pulling together some loose ends on data encoding and sector addresses.

Take a look at the section marked 'SECTOR ADDRESS'. You'll notice that it begins with three hexadecimal numbers 'D5 AA 96'. This is called the 'address header'. The 'read-write track sector' (RWTS) subroutine in DOS looks for this particular sequence of bytes to determine the start of a sector. It then knows that the following bytes are the encoded Volume, Track, and Sector numbers for this sector.

Remember our discussion of the two-for-one encoding scheme above? This is where it is used. Each byte <u>pair</u> is an encoded value. The first two bytes are the diskette volume number, the next two are the track number, and the next two are the current sector number. The last two are the checksum bytes for this header which is the exclusive-or of the unencoded Volume, Track and Sector. Immediately following this are three more hexadecimal numbers 'DE AA EB' which mean "end of address header".

Since it is often useful to be able to read what

sector you are looking at (especially in the Copy][Plus Nibble Editor), here is a table of the encoded and unencoded sector numbers, along with the DOS 3.3 hard and soft sector translation:

Soft Sector	Hard Sector	Encoded Byte pair
0	0	AA AA
7	1	AA AB
E	2	AB AA
6	3	AB AB
D	4	AA AE
5	5	AA AF
C	6	AB AE
4	7	AB AF
B	8	AE AA
3	9	AE AB
A	A	AF AA
2	B	AF AB
9	C	AE AE
1	D	AE AF
8	E	AF AE
F	F	AF AF

In DOS 3.2 and earlier versions, the sectors were written to the disk in "soft sector" order. With DOS 3.3, sectors are written to the disk in ascending order, but translated by RWTS into a soft sector. This was done to maximize sector access time. All you really need to remember is if the disk you are looking at is DOS 3.2, then the sector number you see is the sector number DOS would give you. If you are looking at a DOS 3.3 disk, you must translate the sector number you see (the hard sector) to the soft sector number used by DOS.

We've covered a lot so far, but one important, perhaps the most fundamental part of disk interfacing, has been left unanswered. Since information is written out serially to the drive, when reading it back, how do we know we are "in-sync" with the incoming bits? Perhaps an example would best illustrate this. The following is a sample bit stream that could come off the disk:

1 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 1 0 0 1 0 1 1 0

Those of you who are good at converting binary to hexadecimal will recognize this as the DOS 3.3 address header, ´D5 AA 96´. If we were to start reading from the disk at the beginning of this bit stream, everything would read fine. But what would happen if we started reading one bit late? Or 2 bits late? It´s clear that the hardware only has a 1 in 8 chance of actually reading the data in sync with the way it was written. There are several methods currently employed by disk drive manufacturers to make sure the hardware is in sync with data on the disk. Let´s take a look at some of them, then we´ll show you how the Apple does it.

Many disk drives - Apple drives excluded - have a small LED (light-emitting diode) that shines through a small hole (the index hole) on a diskette. Pick up your Copy][Plus diskette if it´s handy and you´ll notice right next to the large center hole, a small one that most likely doesn´t go through the media on the diskette. If you were to spin (carefully!) the media inside the disk jacket, you would notice at least one place where a small hole would line up with this one and you could see right through. With many disk drives, this is exactly what happens. As the diskette spins, the LED would shine through this hole and an electronic device sensitive to light could detect the precise point at which they align. The hardware can then use this as a

reference point and say ´a sector starts here´.

Well, the creator of the Apple (being somewhat prone to doing things a bit differently from everyone else) decided this just would not do, and he used something known as "self-sync" to determine a sector boundary. If you look back at the sector picture, you will notice several ´FF´ bytes which are labelled ´SELF-SYNC FIELD´ to the right. These are special bytes that, as we´ll see in a moment, have the unique property of being able to make sure the hardware is ´in sync´ with the data on the diskette.

Ordinarily, information is written out to the diskette as eight bit bytes, with the high bit always set. Self-sync bytes, however, are written out as either nine or ten bit bytes. This is done by introducing a short delay after the byte is written equal to one or two bits. When two bits are used, the hardware will always be in sync after five self-sync bytes are read. Self-sync bytes are written to the disk just prior to writing data, and guarantee that the data will be in sync with the hardware when it is read. DOS 3.2 used nine bit self sync bytes since its hardware could not reliably read two consecutive zeros. DOS 3.3 uses ten bit self-sync bytes. When nine bit self-sync bytes are written, larger sync fields are required to guarantee the data is in sync. This is why DOS 3.2 uses such large sync fields, compared to DOS 3.3.

That about raps up our introduction to standard Apple disk recording techniques. Now let´s take a look at how these techniques are modified to produce protected diskettes.

Diskette Protection Schemes

In the last section, you were introduced to standard Apple disk recording methods. Now we´ll show you how they are commonly changed for software protection. We´ll start off with easy changes and progress to the state of the art schemes currently in use.

Changed Address Headers

As you saw in the last section, Apple DOS uses address headers to mark the beginning of a sector. For DOS 3.2 this header is ´D5 AA B5´ and for DOS 3.3, it is ´D5 AA 96´. In addition, there are end marks following both the address and data portions of a sector which are ´DE AA EB´ (sometimes the ´EB´ is not used). Since DOS looks for these, simply changing these to new values will cause any copy program based on DOS (such as COPYA on your DOS master diskette) to fail. This was one of the first and most simple disk protection schemes developed for the Apple but, even today, most disks employ this as one of their protection methods.

Changed Self-Sync

About this time, the first bit copiers appeared. (Remember Copy][Plus 1.0?) They didn´t look for address headers at all. Instead, they looked for the large self-sync fields and determined that a track started right after one of these. It didn´t take long before many copy-protected diskettes used both changed address headers and changed self-sync. One of the most popular values to use was ´FE´ instead of ´FF´. The bit copiers responded by allowing a range of values for the self-sync, and usually defaulted the range to

include both ´FE´ and ´FF´. Next, the sync fields were written with invalid bytes (bytes with more than two consecutive zeros), with only the minimum self-sync required by the hardware. Now bit copy programs had nothing to use to determine the start of a track. Headers were changed, and the sync fields were too small to reliably use. The solution? You guessed it, the bit copy programs included subroutines to convert illegal bytes to some known quantity, usually self-sync.

About this time, the concept of parameters was introduced to bit copy programs. It became obvious that no single set of algorithms would be able to handle all types of copy protection. The user needed to be able to turn certain routines on or off, and to set the operating values for others.

Synchronized Tracks

Somebody got clever one rainy day and decided the real way to copy protect a disk was to change nothing that is visible on the track, just change the alignment of the tracks themselves. When DOS formats a disk, it is always done with a certain alignment, due to the consistency of the formatting routine. If this alignment were changed, then checked for, the copied disk would not work if a standard or bit copy program were used, since the chance of it having the same alignment as the software company´s production software is small. Soon thereafter, bit copy programs included an option to handle synchronized tracks. They copy not only the data, but whatever track alignment is on the original disk as well. Disks that use this scheme include Applewriter ///, Visicalc ///, Castle Wolfenstein, Dark Forest, Datadex, DB Master and many others.

Half Tracks

This method appeared about the same time as synchronized tracks. The Apple disk drive actually has 70 tracks, not 35. This means that the head has 70 different positions. Unfortunately, due to the width of the read/write head used in the drive, it is not possible to write complete tracks on every track boundary. So DOS actually steps the head twice for every track on the disk, giving the familiar 35 tracks. But since it is possible to position the head to any of the 70 half-tracks, some disks shift the data and start using tracks on half-track boundaries. For example, rather than writing information on tracks 0, 1, 2, 3, ... 22, they might use 0, 1.5, 2.5, 3.5 ... 22.5.

An Extra Track?

The hardware can (on most drives) write one extra track after the last track on the disk. Since a normal copy program doesn't suspect that an extra track exists, it won't try to copy it. This is part of the reason bit copy programs such as Copy][Plus allow you to specify start and end tracks to copy. A disk that uses this scheme is Apple World.

Bit Insertion

Remember self-sync bytes? Those are the ones with extra bits on the end used to ensure data synchronization with the hardware. Well, nothing prevents you from putting an extra bit on the end of other bytes as long as the maximum number of consecutive zeros is not exceeded. Several disks (PFS, Visidex, Visischedule, Visiterm, Visifile, Desktop Plan and many others) use this scheme. Whenever the program must access the disk frequently, this scheme is easy, since it doesn't interfere with any DOS routines. This is why so much business software uses it. Unfortunately, there is no way to tell which bytes have extra bits after them, so you generally have to rely on clever trial and error to know which bytes to write as sync. A good rule of thumb: if the disk copies with a standard copy program, but doesn't boot, try synchronized tracks. If that doesn't work, it probably uses this scheme.

Nibble Counting

As you know, you can adjust the speed of your Apple disk drives (see chapter 1). They normally run about 300 rpm, but this can vary significantly. Some software will write a track and go out and see how many bytes (or nibbles - both terms are used) are on the track. It then writes this count on the disk somewhere and, when the disk boots, this count is compared to the actual number of bytes on the track and if they are equal (or within a specified tolerance), the disk works. However, even very small speed variations will affect the number of nibbles on a track, so it is unlikely that your drives will produce the exact same count as the drive which was used to produce the disk. Bit copy programs respond by varying the nibble count somewhat without adjusting the drive speed. This is still one of the more difficult (but not unbeatable) protection schemes.

This covers the main schemes currently in use. It should be noted that several disks use combinations of the above schemes and that in some cases, the combinations form almost a new scheme in themselves. An example of this follows:

Track Arcing

This is really nothing more than synchronized half-tracks with a twist (so to speak). Remember from our discussion of half-tracks, that the Apple disk drive head was too wide to write complete tracks on every half-track boundary. But this doesn't prevent it from writing a smaller amount of information on each half-track, as long as it won't interfere with the data on adjacent half-tracks. For example, let's say that we are going to write about 1/4 the normal amount of information stored on a track, then step the head to the next half-track and do the same. This process is repeated until all the information is written to the diskette. Since each track is short, it never interferes with the track before or after it. If you try to copy this disk, without synchronizing, the half-track images will overwrite each other, and the copy will not work. Furthermore, due to the way synchronizing has been implemented in bit copy programs, using synchronized tracks would also overwrite the adjacent half-tracks. (Copy][Plus has a special parameter to allow synchronized tracks to be used with track arcing.) To further complicate matters, other protection schemes are often used in conjunction with this one, making several disks out on the market very difficult to copy.

It should be noted that disks using track arcing over an extended range of tracks are difficult to backup on some Apple drives. If you have problems backing up a disk using this scheme (examples are Choplifter, Serpentine, David's Midnight Magic), try reversing your drives (copy from B to A instead of A to B).

```
--------------------------------------------
NIBBLE EDITOR     TRACK DUMP    TRACK 08
--------------------------------------------

3340-FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
3350-FFFFFFD5 AA96FFFE AEAAAEAB FFFFDEAA
3360-EBFDFFFF FFFFFFFF FFFFFFD5 AAAD9696
3370-96969696 96969696 96969696 96969696
3380-96969696 96969696 96969696 96969696
3390-96969696 96969696 96969696 96969696
33A0-96969696 96969696 96969696 96969696
33B0-96969696 96969696 96969696 96969696
33C0-96969696 96969696 96969696 96969696
33D0-96969696 96969696 96969696 96969696
33E0-96969696 96969696 96969696 96969696
33F0-96969696 96969696 96969696 96969696
3400-96969696 96969696 96969696 96969696
3410-96969696 96969696 96969696 96969696
3420-96969696 96969696 96969696 96969696
```

This is from a Visidex diskette by Visicorp. Notice the extra 'FD' after the 'DE AA EB' sequence. The 'EB' is actually written as a 10-bit byte. This disk uses the 'Bit Insertion' protection scheme. It also writes invalid nibbles in the sync fields and contains more nibbles than will fit on a disk running at the proper speed. Copy II Plus converts the invalid nibbles to self-sync, then shortens the track so it will fit without adjusting drive speed.

Note that a backup of Visidex can be made using Copy Disk from the main Copy][Plus menu.

```
----------------------------------------
NIBBLE EDITOR     TRACK DUMP    TRACK 01
----------------------------------------

37D0-CCFFFFFF FEFFFFFF FFFFFFCC FFD3ABFF
37E0-FFFFFFFF CCCCFFFF FFFFCA9C A5FFFFFF
37F0-CCCCCCCC CCCCFFFF FFFFFFFF D9D9D9CC
3800-CCCCFFA6 FFFFD2FF FFFFFFFF FFFFFFFF
3810-FFFFFFFF FFFFFFFF F4F4F4F4 FFFFFFD9
3820-FFFFFFD9 FFFFACFF FFFFCCFF FFFFFFFF
3830-FFB7B7B7 FFFF9DFF FFFFFFFF FFFFFFFF
3840-FFFFFFFF F5F5F7FF FFCFFF9A FFEAEAEA
3850-CCFFFFFF FFFFFFFF ACFFFFFF FFFFFFFF
3860-FFFFFFFF FFE9FFFF FFFFFFFF FFFFFFA6
3870-A6A6F6F6 FFB4B4B4 FFFF9696 96FFFFFF
3880-FFFFFFFF FFFFFFFF FFFF939E 9EC9F5FF
3890-F3FFFFFF FFECFFFF CEFECCCC A6A6A6FF
38A0-FFFFFFFF FFFFFFFF FFFFFFFF FFFFDAE6
38B0-E6FFCCCC CCCCCCA6 B9C9C9C9 FFFFCDFF
38C0-FFFFFFFF FFFFFFFF FFFFD4D4 FFFFFFFF
```

This is an unused track (track 1) from Visicalc by Visicorp. Notice that there is no apparent order and no real data is present.

This usually results in an error 2, and it can be ignored when making a backup of this disk.

```
----------------------------------------
NIBBLE EDITOR     TRACK DUMP    TRACK 05
----------------------------------------

3BE0-A0A48881 808080A2 82828585 A082828A
3BF0-FFFFFFFF D5AA96FF FEAAAFAA ABFFFADE
3C00-AAEBC8C8 C8C891A3 FCFFFFFF D5AAADDD
3C10-96F5EEDA EBDFA7AE 9DBAE5ED A79AEBB5
3C20-F49AE9B7 F3DBFABC FCF9E7DE FCF6FAFD
3C30-F6F7E7AE CBBAB6F6 ACB5D7B6 FAB6ACEB
3C40-F5F6CECB B5D6A7F3 FC9DFDDC FEBCA7DE
3C50-AC9E9BFE F2FAFBDA F4B5F3CF FBB7FFFA
3C60-ADB9DBCB 9BDEEEF2 96EDED96 96979797
3C70-979B9B96 96969696 96EAEA97 E5E99DDE
3C80-DDE9EA96 F9D3A7EC E9EA9697 DEAFE796
3C90-969A9AB2 B2BABAB3 B3AEAE9A 9A9D9DF7
3CA0-F79797EA EA9696DD B49EEFF5 A7A7ABB7
3CB0-EA97FFAE B7ABB6AF A7BCB7AE D3E9F6EE
3CC0-F4D3DF96 ECECEEBE B9ACF5ED BDECEDD3
```

This is from a recent PFS diskette by Software Publishing Corp. Notice the invalid bytes in the sync field. Like the Visiseries from Visicorp, this disk also uses a ´Bit Insertion´ protection scheme, but the invalid bytes cause the hardware to unlock, and quite often the sequence cannot be found by the Bit Copy program. The standard copy program in Copy II Plus (see chapter one) has been set up to copy PFS, so you don´t need to worry about setting parameters in the Bit Copy program.

```
------------------------------------------
NIBBLE EDITOR     TRACK DUMP   TRACK 04
------------------------------------------

41D0-FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
41E0-FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
41F0-FFFFFFFF FFFFFFD5 AAB5AAAB AAAEAEAE
4200-AEABDEAA EBFFFFFF FFFFFFFF FFFFFFFF
4210-FFFFFFD5 AAADABAD EBABFFFF FFFFFFFE
4220-EDEBFFFF FEADFFFF BEABABAB ABABABAB
4230-ABFBBEAF FFFFFFFF FFEBBEEB ABABEDFF
4240-FFEDABAB ABABABAB ABABADAD BEFFFFFF
4250-FFFFFFB5 B5FFFFFE ADFFFEEB ABABABAB
4260-ABABABAB ADABBFFF FFFFFFFF FBADBBAB
4270-ABBEFFFF FEEBABAB ABABABAB ABADDDB5
4280-FFFFFFFF FFFEEDEB FFFFFEAD FFFFBEAB
4290-ABABABAB ABABABAE ABEFFFFF FFFFFFFB
42A0-AEEBABAB EDFFFFFD BBABABAB ABABABAB
42B0-ABABABAB ABABABAB B6DBDBDB BEABABF7
42C0-ABDBDBDB DBDBDBAB F7ABABAB ABABABAB
42D0-ABABEDFF DBDDDBDB DBBEBEDB DBDBDBDB
```

This is from a normal 3.2 diskette. Notice the large self-sync gaps, and the 'D5 AA B5' address header.

```
------------------------------------------
NIBBLE EDITOR     TRACK DUMP   TRACK 04
------------------------------------------

3030-FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
3040-FFFFFFFF FFFFFFFF FFFFFFFF FFFFD5BF
3050-B5FFFEAA AEAAAFFF EADEAAEB FFC6DFF7
3060-FBFDFEFF FFFFFFFF FFFFD5AA ADABABAB
3070-ABABABAB ABABABAB ABABABAB ABABABAB
3080-ABABABAB ABABABAB DDDDABAB ABABABAB
3090-ABABABAB ABABABAB ABABABAB ABABABAB
30A0-ABABABAB ABABABAB ABABABAB ABABABAB
30B0-ABABABAB ABABABAB ABABABAE AEABABAB
30C0-ABABABAB ABABABAB ABABABAB ABABABAB
30D0-ABABABAB ABABABAB ABABABAB ABABABAB
30E0-ABABABAB ABABABAB ABABABAB ABABBEBE
30F0-ABABABAB ABABABAB ABABABAB ABABABAB
3100-ABABABAB ABABABAB ABABABAB ABABABAB
3110-ABABABAB ABABABAB ABABABAB ABABB5BA
3120-AFABABAB ABABABAB ABABABAB ABABABAB
3130-ABABABAB ABABABAB ABABABAB ABABABAB
```

This is from Apple Panic, by Broderbund Software. The address header has been modified to 'D5 BF B5' and there are some glitches in the second sync-field, which the Bit Copy program can either clean up or ignore.

The Bit Copy program can make a backup of this disk by Copying tracks 0 to D. Set parameter 9 to 0 and param 8 to 1.

```
-----------------------------------------
NIBBLE EDITOR     TRACK DUMP    TRACK 00
-----------------------------------------

3CE0-FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
3CF0-FFFFFFFF FFFFFFFF FFFFFFD5 AA96AAAA
3D00-AAAAAAAA AAAADEAA EAFFFFFF FFFFFFFF
3D10-FFD5AAAD EAEAA69F D7F5FFF6 BBF5F3E9
3D20-EBAC979E F6F2E9B2 DBDBD69D AFADBF9A
3D30-CBB5B49A B7E7B6F2 B7E6E697 E7F2CFE7
3D40-E99A9A9A E7B5E9F3 B4D6BFD9 9A9AACA6
3D50-BABABED9 D6B6EAAB ACEFB6E6 ACFBB7D7
3D60-979EF9BF FABFCDF9 D79696A7 E9CBF6E6
3D70-CEBADFDA D7E69AAB D7E6E6D7 D9E7DFD7
3D80-D7D7D7E9 E9DBDBEF 9FACB3EF F6F3B7BD
3D90-DAB5F9F9 F9F3B3B5 E6D7E596 CFE5BCDF
3DA0-EBFDFBBF 96CBBAFD ABF29DE9 CFDC9AE6
3DB0-D79797DF DAE7D6DF DADFDADC AEDA9BD6
3DC0-EDB4BEF4 EDABBC97 ACDFDAE6 D7DCAEDA
```

This is a normal 3.3 diskette. Notice the small size of the sync fields and the address header ´D5 AA 96´.

```
-----------------------------------------
NIBBLE EDITOR     TRACK DUMP    TRACK 01
-----------------------------------------

4300-DFFCFFFF FFFFFFFF FFFFFFFF FFFFFFFF
4310-FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
4320-FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
4330-FF9492CD AAABFAAF BBAFEEEB AEABEAFA
4340-AEBAFAAF BFABAFEA AFEAAFEA AFEAAEAB
4350-EAEAEAAF BFBFFEAB AEFEEAAF BFBEAEBA
4360-FEAFFFFE AEAABEEF FFFFAEAA EEAFFFFE
4370-AEAAFFAE FFFFAEAA BABAAABF FEBFAEEF
4380-AEAAEAAF BEBFEFEE FFFFAEAA FEAFFFFE
4390-AEAAEAAF BBAFEFEE FFFEAEAA FBAEBFAB
43A0-BEEEBFBE AAAAFFEE FFFEAEAA FFEEFFFE
43B0-AEAABAAA EEABFFFE BAAAEBBB FFFEBAAA
43C0-BFAFFFFB FBAEBFAB BEEEFEFF AEAAAAAA
43D0-AEAFAFAB AEABAEEE EEFAAFAF EBBEEBBF
43E0-EFBAEFBB EEBFEFBE EFBFFBAE FBAFFFAB
43F0-FEAEFEAF FFAEFFAF FBBAFBBB FABEFABF
4400-FBBEFBBF FEBBFFBA FFBBFEBE FEBFFFBE
```

This is from track 1 of the Copy II Plus diskette. To make it load fast, no address headers are used, and the entire track contains one sector. Notice the different data header, ´94 92 CD´.

```
---------------------------------------------
NIBBLE EDITOR     TRACK DUMP   TRACK 02
---------------------------------------------


3570-FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
3580-FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
3590-FFFFFFD5 AAB5BBAA BFBFEFEF BFFEDEAA
35A0-EBFFFFFF FFFFFFFF FFFFFFFF FFFFFFD5
35B0-AAADABB7 ADADBDB7 DBDDD7BB B7FBEAFB
35C0-DFB7EBB6 FEF6FFDD BEAFAFFD EEDFFDB5
35D0-DEABABB5 B5EBBDD6 ADD7BBFE BDFEEBF6
35E0-EAF5BEFE FBF7BEB5 B5DFBEF5 BEFEDFAB
35F0-EDFDBDED EBFEDBEA EFFDDEF6 DBBFEAAE
3600-EDD6F5AD DEEAFABB DDBBBDAF AEFFBDFA
3610-B6FBEBD7 BFEEBDFA AEB7DEB6 F5FFBFBE
3620-EEABAFF5 FDB6BADB BDEBFDD6 B7B5DBFA
3630-BDB6AFFB BBD6B6EE B6D7DBDD AFF7FDBA
3640-FABEF6D7 B7FBEFEF FBEBEDAD FAAFFEDF
3650-DEABADBD BDBAADAF BFBFADAF AFAFABAB
```

This is also from Visicalc by Visicorp. Notice how the volume, track, sector and checksum information in the address field is changed from standard. This prevents a normal copy program from copying it, but in no way interferes with the Bit Copy operation.

```
---------------------------------------------
NIBBLE EDITOR     TRACK DUMP   TRACK 01
---------------------------------------------


3660-FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
3670-FFFFFFFF D5AA96FF FEAAABAA AAFFFADE
3680-AAEBFFFF FFFFFFFF FFD5AADB BAABA7DF
3690-D9ECECAB ADF5FBAF BBAEBFE6 B6ABFBB4
36A0-FCDB97D7 DAD79AEB FF97DCDB B5B5B4DC
36B0-9FEDFAE6 FACD9DAE A6CEBBDD DFEBDFA7
36C0-AEAEACD9 DFD6DC9D 9EA7FED9 BDDF9DEB
36D0-EBCDB7DD AEDA9DD9 DEDEE9F7 F7EB96EB
36E0-B9FEEFEE EDB4ABBD B9B3AD9B B797BE96
36F0-BEBEBF97 CBBE9DCE CB9E9E9A 96969ACB
3700-9FABB3BE B3BBCEB4 AFBFB4BC B4AFCBA7
3710-B7BEBD97 969696CB 9ABE9696 CBBE9ACD
3720-9FBFCB96 CBBEA6CF BDABB6AD 97B9D39A
3730-969696BF ADB5BE9B BFCB97BF 96BDBDA6
3740-A69FBFCD BE9ACBCE 9DBEBF9D A69ACB96
```

This is from Applewriter II by Apple Computer, Inc. Notice the changed data header, to ´D5 AA DB´. Changes were also made in the data field to prevent standard copy programs from functioning.

Nibble Editor

Copy][Plus contains a nibble editor which is invaluable for examining disks to determine what protection schemes they use. It can be invoked either during the copy process, or through a special scan disk mode which may be selected upon entry to the Bit Copy option of Copy][Plus.

When you select the Bit Copy Option, your screen should look like:

```
        COPY ][ PLUS BIT COPY PROGRAM
   (C) 1982 CENTRAL POINT SOFTWARE, INC.
  ----------------------------------------
     COPY OR EDIT (C OR E)  DEFAULT=COPY
```

Answering this with ´E´ for edit will cause Copy][Plus to enter the nibble editor after each track is read. Also, since it will not be copying, it won´t ask anything about the duplicate disk.

After the first track is read, you will see a display similar to:

```
        COPY ][ PLUS BIT COPY PROGRAM
   (C) 1982 CENTRAL POINT SOFTWARE, INC.
  --------------------------------------
   TRACK: 0   START: 3000  LENGTH: 47FF

   2FE8: 00 00 00 00 00 00 00 00    VIEW
   2FF0: 00 00 00 00 00 00 00 00
   2FF8: 00 00 00 00 00 00 00 00
   3000: FF FF FF FF FF FF FF FF
   3008: FF FF FF FF FF FF FF FF  <- 300C
   3010: D5 AA 96 FF FE AA AA AA
   3018: AA FF FE DE AA EB FF FF
   3020: FF FF FF FF FF FF FF FF
   3028: FF FF D5 AA AD BF FE DF
  --------------------------------------

    A  TO ANALYZE DATA     ESC TO QUIT
    ?  FOR HELP SCREEN     / CHANGE PARAMS
    Q  FOR NEXT TRACK      SPC TO RE-READ
```

This is the main nibble editor display. The nibble editor actually works only in the middle window, between the title and the option section at the bottom. The first line indicates what track you are currently editing, its start in memory and its length. Since no analysis has been done yet, it is pointing to the start and end of this track´s buffer. Beneath this, on the left, is the actual track image. It is shown as an address followed by 8 hexadecimal data bytes per line. The word "VIEW" to the right lets you know you are in view mode, and can scroll through the track buffer. The address at the right marked by "<-" is the actual memory address you are at in the track´s buffer.

Several options are displayed in the bottom window. You can ask Copy][Plus to perform its track analysis routines (per the current parameter settings) by pressing "A". Pressing "?" will present you with the nibble editor commands, and "Q" will quit this edit and move on to the next track. "ESC" will exit the editor and return you to the sign-on screen, and "SPC" will cause the track to be re-read and return you to the initial edit position. "/" operates just as it does in copy mode, allowing you to change parameters. Using "/", "SPC", and "A" in sequence allows you to view a track, make any parameter changes you wish, then have Copy][Plus analyze the track just as it would if you were copying the disk. This is an invaluable aid in correctly setting parameters.

If you press "?" you will be presented with the Copy][Plus nibble editor help screen, which is really a list of commands. It should look like this:

```
                NIBBLE EDITOR COMMANDS

BEGINNING  B                    C CHANGE NIBBLE
UP 32      T                    F FIND NIBBLE
UP         I                    R REPEAT FIND
LEFT    J      K  RIGHT
DOWN       M                    S TOGGLE SYNC
DOWN 32    V                    Q NEXT   TRACK
END        E                    ? HELP
RESET BEG  CTRL-B               P PRINT TRACK
RESET END  CTRL-E             RTN RETURN TO EDIT
```

The cursor commands are quite straightforward, and allow you to move anywhere within the track buffer with a minimum of effort. The options ´RESET BEG´ and ´RESET END´ are useful in calculating a track´s length, and in the copy mode will actually reset the track beginning and end for the write routines. ´C´ allows you to change nibbles, and you will notice the ´VIEW´ status becomes ´CHANGE´

when ´C´ is pressed. You may enter any string of hex nibbles separated by spaces and they will be placed at the current cursor position. ´F´ allows you to find a string of nibbles. You will see the prompt ´FIND´ appear in the lower right of the nibble edit display and you can enter in any 1 to 3 nibble sequence for the editor to find. Spaces are optional. If the string is found, you will be placed at its beginning. If it is not, you will be placed at the end of the track buffer. ´R´ is useful to repeat the find command for the last specified string. ´S´ will toggle the sync flag for the nibble at the cursor position, converting standard bytes to sync, and sync bytes to standard. ´P´ allows you to print a track. It will start printing at the current cursor location and extend to the end of the buffer if no analysis has been done, or to the track end if analysis has been performed. The slot number, page length, and printer line feed character are Copy][Plus parameters and may be changed at any time.

When using the editor from copy mode rather than from edit, the ´A´ to analyze, ´/´ to change parameters, and "SPC" to re-read commands are not available, as they apply only to edit mode. See the "DISPLAY" parameter in the Copy][Plus parameter list for instructions on allowing edit from copy mode.

We encourage you to practice using the editor on a standard disk to become familiar with its capabilities and also to view the formatting discussed in earlier sections. The nibble editor is your best tool for making backups of disks not on the instruction list supplied with this manual.

Copy][Plus Parameters

By now, you are undoubtedly ready to hear more about the internal Copy][Plus routines and their parameters, and that is the focus of this section. We´ll start by introducing the various internal routines and present each of the parameters that affect their operation.

These first routines are those that clean the sector data and set the sync fields. They are performed in edit mode prior to track analysis, so you may need to disable them for diskettes that are radically different from standard formats.

>>> READ A TRACK. This routine simply reads nibbles from the original drive until it fills the track buffer. The only parameter associated with this routine is the buffer size which is controlled by the number of track read buffers (BUFFS). This is Copy][Plus parameter 3C, which can take on values of 1, 2, or 4, where 2 is the default value. This is the number of tracks Copy][Plus will read and write per pass. Single drive users can use a value of ´4´ for this parameter which will reduce the amount of diskette swapping they must do. However, this will also limit the analysis that can be performed, and Copy][Plus may not be able to copy certain disks in this mode.

>>> FIX INVALID NIBBLES. This routine scans the input buffer for occurrences of invalid nibbles. These are nibbles that the hardware cannot read reliably (those with more than two consecutive zeros). It will replace any invalid nibbles with standardized sync nibbles. Its controlling parameter is number 31. A ´1´ turns this routine on, a ´0´ disables it. The default value is ´1´ (on). These are the nibbles it will convert to self-sync:

80 81 82 83 84 85 86 87 88 89
8A 8B 8C 8D 8E 8F 90 91 98 A0
A1 A2 A3 B0 B1 B8 C0 C1 C2 C3
C4 C5 C6 C7 C8 D0 D1 D8 E0 E1
E2 E3 E8 F0 F1 F8

>>> CLEAN SYNC FIELDS. This routine scans the track buffer for sync fields and sets all values within the field to standardized self-sync, including any glitch bytes or data present. It looks for the address and data epilogue nibbles (usually DE AA XX, but modifiable through parameters 19, 1A and 1B) and converts all values to self-sync until it finds an address or data prologue (usually D5 AA, but modifiable through parameters E, F, 1C and 1D). This routine is enabled/disabled through parameter 9. A ´1´ turns it on, a ´0´ turns it off. Default value is ´1´, enabled. Note: This routine needs to be disabled for many protected diskettes, as it has the potential to "clean" many valid data nibbles.

>>> STANDARDIZE SELF-SYNC. This routine scans the track buffer for sync fields and sets bytes within the field to standardized self-sync. It uses the following parameters:

PARAMETER NAME	PARAMETER NUMBER
SYNC.MIN	4
SYNC.MAX	5
SYNC.#	6
GLITCH.SIZ	32
CHANGE	33
STAND	7

SYNC.MIN is the minimum value for self-sync bytes;

SYNC.MAX is the maximum value. SYNC# is the minimum number of self-sync that must be found to constitute a valid sync field. GLITCH.SIZ is the number of non self-sync bytes that can be found within the field. CHANGE is a flag that determines whether or not to convert the non-sync bytes to self-sync (1=yes). STAND is the standardized self-sync value and usually has its high bit clear so the write routine will write the value as self-sync. This routine is controlled by parameter 8 which if it is a ´1´ enables this routine, or if it is a ´0´, disables it.

The next two routines are used to find the start of a track. Tracks may be found either by ´sync´ or ´header´. Copy][Plus will always try to find a header first, and if this fails, it will look for sync.

>>> FIND HEADER. This routine looks for an address header to determine the track start. The parameters it uses are:

PARAMETER NAME	PARAMETER NUMBER
MATCH	A
HEADER.TABLE	E

MATCH is the number of values that must match the headers in the header table. HEADER.TABLE is the actual table of all address and data prologues and epilogues. These are broken down as follows:

ADDRESS.START	E TO 18
ADDRESS.END	19 TO 1B
DATA.START	1C TO 1E
DATA.END	1F TO 21

The ADDRESS.START sub-table contains 3 bytes for the address prologue, and 8 bytes for the encoded volume, track, sector and checksum. ADDRESS.END, DATA.START and DATA.END each contain 3 bytes. A zero byte will match any value.

This routine is always enabled and, upon failure, the next routine, FIND SYNC, will be called. If it also fails, a read error will occur.

>>> FIND SYNC. This routine will attempt to find the track start by looking for the largest sync gap in the first part of the track buffer. If no sync fields are found, a read error occurs. The track start is set as the first byte beyond the sync field. Its parameters are: SYNC.MIN (4), SYNC.MAX (5) and SYNC.# (6). Note that if BUFFS is set to 4, this routine will look only for the first sync gap, not the largest.

>>> FIND END. This routine finds the track end. It looks for a duplicate image of the track start, then moves back over the last sync field or other garbage that may be present. The user can also specify a track length and this will be added to the track start to determine the track end. The parameters for this routine are:

PARAMETER NAME	PARAMETER NUMBER
END.MATCH	50
END.GLITCH	3A
TRACK.GAP	3B
CUT?	44
CUT.HIGH	45
CUT.LOW	46

END.MATCH is the number of bytes to match with the track start to determine if this is really the end of the track. Once the end of track is found, END.GLITCH is the maximum number of non self-sync bytes that can be found in the last sync field.

TRACK.GAP is a value to subtract from the track end to shorten tracks that contain many non-standard bytes at the end. CUT? is a flag that, when set, tells Copy][Plus to forget looking for an end and just cut the track off CUT.HIGH and CUT.LOW bytes from the track start.

This routine is always enabled. A value of ´0´ in TRACK.GAP and CUT? will cause Copy][Plus to find the end itself and back up over the last sync field.

>>> BIT-INSERT. This routine will look for a sequence of bytes that match those in the bit-insertion table. The values in the table can have their high bits cleared, which indicates they are to be written as self-sync. The pattern matching with the track buffer ignores high bits. When a match is found, the corresponding high bits in the track buffer are also cleared, which will force the write routine to write them as 9 or 10 bit bytes, rather than 8 bit bytes. The parameters used by this routine are:

PARAMETER NAME	PARAMETER NUMBER
BIT.FLAG	34
BIT.TABLE	35
second byte	36
third byte	37
fourth byte	38
fifth byte	39

If BIT.FLAG is set, this routine is enabled. Any zero values in the bit insertion table are wildcard values and will match anything. Remember to clear the high bit of any value that is to be written as a 9 or 10 bit byte.

>>> READ.VERIFY. This routine will verify the

track just read. It is only enabled when the number of read buffers (BUFFS, param 3C) is set to ´1´. If BUFFS is set to ´1´ and read verify is to be disabled anyway, parameter 41 may be set to ´1´. Copy][Plus will compare the first two track images in memory, and if they are not identical (except for sync fields), a read error will be produced, and Copy][Plus will re-read the track. Parameter 42 determines how far ahead to look for self-sync when a match error is found. If self-sync is found within this range, the error is ignored.

>>> WRITE TRACK. This routine writes the track in memory to the duplicate disk drive. It writes all self-sync as either 9 or 10 bit bytes depending upon the value of BITS, parameter 3E. If BITS is set to ´1´, 9 bit bytes will be written, if set to ´2´, 10 bit bytes are selected.

>>> NIBBLE COUNTING. This routine computes the number of nibbles on the original diskette and tries to maintain that count on the duplicate disk. It is called after the WRITE TRACK routine. It works by converting normal bytes to 9 or 10 bit bytes if the track is too short, or by converting self-sync to 8 bit bytes if it is too long. It calculates the number of bytes to convert based upon the difference between the length of the original and duplicate tracks and by the current setting of BITS. The difference between the two counts is compared to TOLERANCE (parameter 4C) and if it is less than or equal to this number, no error is produced; otherwise a nibble count error (error 9) is displayed. This routine is enabled by setting DONIB? (parameter 4B) to ´1´.

>>> WRITE VERIFY. This routine verifies that the track just written is correct. Two levels of write verification are selectable. The default mode

simply checks that the track start was not over-written by the track end (track too long error). A more comprehensive write verify may be selected which will re-read the track, analyze it, and compare it byte for byte to the original track. This level is only available if BUFFS (the number of read buffers) is set to ´1´. If BUFFS is set to ´1´ and extended write verify is not desired, it may be de-selected by setting parameter 47 to ´0´.

>>> SYNCHRONIZE TRACKS. This routine makes sure that the tracks on the duplicate disk have the same alignment as those on the original diskette. It is controlled by several parameters:

PARAMETER NAME	PARAMETER NUMBER
DOSYNC	D
SYNC.TRACK	C
SYNC.START	22
SYNC.MATCH	30
ARC	3F

DOSYNC is simply a flag to enable or disable track synchronization. SYNC.TRACK is the track to use as a reference for synchronization. SYNC.START is a table of bytes to match for the start of the SYNC.TRACK, and is identical in structure to the header table for normal tracks. SYNC.MATCH is the number of bytes in the table to match. ARC is a special parameter for track arcing that forces Copy][Plus to do only a minimal lead-in when writing the track image. This allows the use of synchronized tracks with half-tracks written on each half-track boundary. Because the lead-in is very small, this should not be used with standard diskettes.

>>> TRACK CHOPPING. This routine will chop a track that is too long by shortening all the sync fields to a length specified in KEEP (parameter 3d). It is called automatically by WRITE TRACK if a track too long error occurrs.

>>> DYNAMIC HEADER CHANGE. Some disks change the address header for each track on the disk. They usually store the new header at the end of the current track. Using this routine, you can tell the Bit Copy program where to find the new header and it will dynamically update the address header table. It uses these parameters:

param number	description
51	Flag to turn on this routine (1=on).
52	Low byte of where to find new header.
53	High byte of where to find new header.
54	Fill header backward (1) or forward (0).

The new header is found by adding the address in parameters 52 and 53 to the start of the track. Parameter A is used to determine the length of the header (number of bytes to fill into the header table.)

The following are miscellaneous parameters not directly tied to a given routine in Copy][Plus:

>>> EREAD (0) is the number of read retries.
>>> EWRITE (1) is the number of write retries.
>>> TRKMAX (2) is the maximum track length.
>>> TRKMIN (3) is the minimum track length.
>>> DISPLY (B) sets the display type:
0 = off.
1 = on.
2 = on with editor entry.

Use B=2 to enter the nibble editor while doing a copy after each track is read.

>>> HEADLY (40) sets the head step delay.
>>> PRSLOT (48) sets the printer slot.
>>> PRLF (49) sets the line feed character.
>>> PRLINE (4A) number of printed lines/page.
>>> SHIFT (4D) # of bytes to shift track start and end.
>>> REDERR (4E) reduced error check on read.
>>> SDFLTR (4F) sets a read error if any bytes with more than 1 consecutive 0.
>>> EMATCH (50) number of bytes to determine a track end.

That's all of them. In Appendix A you'll find a numerical list of these parameters, their name and their default values to use as a reference.

Bit Copy Parameters

Here is the list of the Copy][Plus Bit Copy program parameters in numerical order. Each parameter is given a name for reference and the default value is also shown.

PARAMETER #	PARAMETER NAME	DEFAULT VALUE
0	EREAD	3
1	EWRITE	3
2	TRKMAX	1A
3	TRKMIN	10
4	SYNC.MIN	FE
5	SYNC.MAX	FF
6	SYNC.#	8
7	STAND	7F
8	STANDF	0
9	CLEAN?	1
A	MATCH	9
B	DISPLAY	1
C	SYNC.TRACK	0
D	DOSYNC	0
E	ADDRESS.START	D5
	...	
19	ADDRESS.END	DE
	...	
1C	DATA.START	D5
	...	
1F	DATA.END	DE
22	SYNC.START	D5
	...	
30	SYNC.MATCH	9
31	FIX.INVALID?	1
32	GLITCH.SIZ	2
33	CHANGE	1
34	BIT.FLAG	0
35	BIT.TABLE	DE
	...	
3A	END.GLITCH	2

BIT COPY PARAMETERS

3B	TRACK.GAP	0
3C	BUFFS	2
3D	KEEP	11
3E	BITS	1
3F	ARC	0
40	HEADLY	40
41	OVERIDE.VER	0
42	SYNC.LOOK	B
43	PAGE.OVF	1
44	CUT?	0
45	CUT.HIGH	8
46	CUT.LOW	1F
47	IG.WRITE.VER	1
48	PRSLOT	1
49	PRLF	0
4A	PLINE	3B
4B	DONIB?	0
4C	TOLERANCE	1
4D	SHIFT	0
4E	REDERR	0
4F	SDFLTR	0
50	EMATCH	9
51	DYNAM	0
52	DYNAM.LOW	07
53	DYNAM.HIGH	08
54	FILL.ORDER	1

Decimal to Hex Conversion Table

Decimal	Hex	Decimal	Hex
0	0	19	13
1	1	20	14
2	2	21	15
3	3	22	16
4	4	23	17
5	5	24	18
6	6	25	19
7	7	26	1A
8	8	27	1B
9	9	28	1C
10	A	29	1D
11	B	30	1E
12	C	31	1F
13	D	32	20
14	E	33	21
15	F	34	22
16	10	35	23
17	11	36	24
18	12	37	25

Copy II Plus 4.3 Disk-Backup Instructions 3/28/83

'T' indicates a track number. When a range of tracks are to be copied, you will see "Tx-Txx". This means set the start track to "x", and the end track to "xx". f only a single track is to be copied, you will see "Tx". This means use "x" or both start and end tracks. Numbers to the right are parameter changes that should be made before copying the tracks shown. "STEP" means track increment.

When making a backup, be sure to follow the steps in order. Often a parameter will not be re-listed if it is set for a prior range of tracks.

If a parameter listing includes "Sector Edit", use the Copy II Plus sector editor to modify the track and sector shown. Be sure to patch the read/write routines if the listing shows "Patched" and to use the correct DOS (3.2 or 3.3).

Some diskettes can be duplicated using the default parameters (or Copy Disk from the main menu). If the diskette you wish to backup is not listed, try the default settings or Copy Disk first.

A "*" next to the product name indicates these parameters were user submitted and have not been verified by Central Point Software. We encourage our customers to let us know when they backup a disk not on this list. This information is made available to all Copy II Plus owners.

NOTE TO ALL IBM PC OWNERS: Copy II PC is now available. Same price as Copy II Plus for the Apple, and it is hands-down the fastest, most reliable and most powerful copy program for the Personal Computer.

```
3-D GRAPHICS SYSTEM *   (Calif Pacific Computer)
        T0-T8
        T11-T12
        T15-T17

3-D GRAPHICS SYSTEM *
        T0-T2
        T4-T8
        T11-T18

ADVENTURE TO ATLANTIS * (Synergistics)
        T0-T22          10=96, 24=96, 9=0, 31=0, D=1

AIR SIMULATOR *         (Mind Systems)
        T0-TF

AIR TRAFFIC CONTROLLER *
        T0-T22          10=96
        T23             31=0, 50=1, 10=96

AKALABETH *
        T0              9=0, 31=0
        T2-T3           E=DE, F=AA, 10=AD
        T6-T18

ALIEN RAIN & TYPHOON    (Broderbund)
        T0-T5           9=0, 31=0, D=D5, F=0
        T6-TE           E=DE

ALKEM STONES *
        T0-T22          A=3, 10=96
```

APPLE ADVENTURE *
T0-T22 D=1, 10=96, 24=96

APPLE CILLIN II *
T0-TC

APPLE //e BUSINESS GRAPHICS *
T0-T22 D=1, 10=96, 24=96

APPLE /// BUSINESS GRAPHICS *
T0-T22 (ERROR 2 OK)

APPLE LOGO *
T0=T22
T1 A=1, 4B=1, 50=1, E=FC, 19=FD, 1C=AA, 1F=EE

APPLE LOGO * (ALTERNATE)
T0=T22
T1 A=1, 4B=1, 50=1, E=AA, 1C=AA

APPLE LOGO * (ALTERNATE)
T0-T22
T1 A=1, 4B=1, 50=1, 3B=1, 4D=8

NOTE: We have been told that Apple Logo requires persistence! Keep retrying track 1 until the disk works.

APPLE PANIC * (Broderbund)
T0-TD

APPLE PANIC *
T0-T5 9=0, F=0
T6-TD E=DE

APPLE PILOT AND SUPER PILOT *
T0-T22

APPLE WORLD * (USA)
T0-T23

APPLEWRITER II (AND e) (Apple)
T0-T22 10=96

APPLEWRITER II PRE-BOOT *
T0-T22 10=96, 9=0

APPLEWRITER /// (Apple)
T0-T22 D=1, 10=96, 24=96

A2-PB1 (PINBALL) (Sub Logic)
T0 10=96
T1-T15 A=3, E=DB, F=AB, 10=BF, 44=1, 45=D, 46=F

AZTEC *
T0-22 D=1, 10=96, 24=96

BACK-IT-UP II * (Sensible)
T0 10=96, 9=0
T1.5-TB.5 10=B5, A=3

```
BATTLE OF SHILO *
        T0-T22            E=D4, 10=B7

BEER RUN
        T0                9=0
        T1.5-TD.5         D=1, 3B=40

BOMB ALLEY *
        T0-T22            E=D4, 10=B7, 34=1, 37=6E, 38=FE

BORG *                    (Sirius)
        T0                10=96, 9=0
        T1.5-TB.5         D=1, 24=96, A=3, E=DD, F=AD, 10=DA, 3B=40
        TD-T20

CANNONBALL BLITZ *
        T0-T22
        T3-TF             3B=1, A=1, 4B=1, 4D=8, 50=1  (ERROR 6 OK)

CANNONBALL BLITZ (alternate)
        T0-T22            10=96
        SECTOR EDIT DOS 3.3 PATCHED
          TRACK 17, SECTOR E.
          CHANGE ADDRESS CD FROM 49 TO 60

CASTLE WOLFENSTEIN        (Muse)
        T0-T22            D=1, 31=0

CAVES OF OLYMPUS *
        T0-T22            10=96, 9=0

CEILING ZERO *
        T0-T2
        T3-T11            9=0, E=D6, 1C=D6, 34=1, 38=F9, 4F=1

CHESS 7.0 *               (Odesta)
        T0-T22            10=96, 9=0

CHESS 7.0 *
        T0-T22            10=96, 9=0, 8=1, 3E=2

CHOPLIFTER & SERPENTINE (Broderbund)
        T0                A=3, 44=1, 45=D, 9=0, 0=F, 50=3
        T1-T8             4=FD, 31=0, 43=0, 45=10, 4F=1, 46=12
        T9                45=8, 46=D
        TA-TB             45=2
        TC-T1E.5 STEP .5        45=8, 10=D4, 51=1, D=1
        T20               45=6, D=0, 4F=0
```

Note: Choplifter, Serpentine, David's Midnight Magic and Starblazer use track arcing and are very sensitive to drive speed. If you have problems, try reversing drives.

```
COLOSSAL CAVE ADVENTURE *
        T0-T22

CONGO *
        T0-T22            D=1, 9=0, 24=96, 10=96
```

```
CRANSTON MANOR           (On-Line)
        T0-T22
        T18              3B=1, A=1, 4B=1, 4D=8, 50=1  (ERROR 6 OK)

CROSSFIRE                (On-line)
        T0-TB            9=0
        T1               3B=1, A=1, 4B=1, 4D=8, 50=1   (ERROR 6 OK)

CRUSH, CRUMBLE AND CHOMP *
        T0-T22           10=96, 9=0

DARK CRYSTAL *
        COPY DISK FROM MAIN MENU ALL 4 DISKS
        SECTOR EDIT DISK 1A AS FOLLOWS:
        TRACK 5, SECTOR F, CHANGE ADDRESSES A8-AA ALL TO EA
        TRACK 7, SECTOR C, CHANGE ADDRESSES 22-24 ALL TO EA

DAVID'S MIDNIGHT MAGIC  (Broderbund)
        T0               A=3, 44=1, 45=D, 9=0, 0=F, 50=3
        T1-TA            44=0
        TB               44=1, 31=0, 43=0, 45=8
        TC-T19 STEP .5   10=F5, F=FD, 51=1, 4F=1, D=1
        (See notes for Choplifter)

DB MASTER * (OLD)        (Stoneware)
        T0-T5            10=96, 24=96, D=1
        T6.5-T22.5       D=0

DEADLINE *               (Infocom)
        T0-T22

DESKTOP PLAN II          (Visicorp)
        T0-T22           10=96, 34=1, 36=2A

DISK ORGANIZER *
        T0
        T1               3B=1, A=1, 4B=1, 4D=8, 50=1 (ERROR 6 OK)
        T2-T4            D=1
        TA-TB

DLM SOFTWARE *
        T0-T22

DRAGON FIRE *            (Level-10)
        T0-T22

DUNGEON *                (TSR Hobbies)
        T0-T22           10=96, 9=0

EARLY GAMES *
        Use Copy Disk from Main Menu

EDUCATION ACTIVITIES SOFTWARE *
        T0-T22

EINSTEIN COMPUTER *
        COPY DISK FROM MAIN MENU
        SECTOR EDIT TRACK 8, SECTOR 4
        CHANGE ADDRESSES 2A-2C FROM BD 8C C0 TO 4C E2 91
```

ELECTRIC DUET * (Insoft)
Use Copy Disk from Main Menu

ESCAPE *
T0-T22

EXECUTIVE SECRETARY *
T0-T22 9=0, 8=1, 10=96

EXPEDITOR (On-Line)
T0-22 10=96
T3 & T1F 3B=1, A=1, 4B=1, 4D=8, 50=1 (ERROR 6 OK)

FIRST CLASS MAIL * (Continental)
T0-T22

FORMAT II *
USE COPY DISK FROM MAIN MENU

FORMAT II (VERSION 7) *
T0-T22 10=96
SECTOR EDIT DOS 3.3 TRACK B SECTOR 5
CHANGE ADDRESSES 4-6 FROM A9 03 BD TO 4C 31 68

FS-1 (FLIGHT SIMULATOR) (Sub Logic)
T0 10=96
T1.5-T21 STEP 1.5 E=DB, F=AB, 10=BF, A=3, 4E=1
T7-T8
T9.5

GALACTIC GLADIATORS *
T0-T20 10=B7, E=D7, 9=0, 31=0
T21-T22 34=1

GAME SHOW * (C.A.I.)
T0-T22 9=0

GENERAL MANAGER * (Sierra)
Working program & sample files use Copy Disk from main menu
Master program:
T0-T22 9=0

GOBBLER * (Sierra)
T0-T22 9=0
T3 3B=1, A=1, 4B=1, 4D=8, 50=1 (ERROR 6 OK)

GORGON * (Sirius)
T0 10=96, 9=0
T1.5-E.5 D=1, 24=96, A=3, E=DD, F=AD, 10=DA, 3B=40

GRAPHTRIX *
T0-T22

HADRON * (Sirius)
T0 10=96, 9=0
T1.5-TE.5 D=1, 24=96, A=3, E=DD, F=AD, 10=DA, 3B=40

HELLFIRE WARRIOR *
T0-T22

HI-RES COMPUTER GOLF *
T0-T22 (both sides)

HI-RES FOOTBALL
T0-T22

HOME ACCOUNTANT *
T0-T22 9=0, 10=96

HYPERSPACE WARS * (Continental)
T0-T22 9=0

INVOICE FACTORY *
T0-T22

JAW BREAKER * (On-Line)
T0-T22 9=0
T3 3B=1, A=1, 4B=1, 4D=8, 50=1 (ERROR 6 OK)

KNIGHTS OF DIAMONDS (BOTH SIDES) *
T0-T22 10=96, 24=96, D=1
MAKE SURE DISK IS WRITE PROTECTED BEFORE USING

KRELL LOGO *
T0-T22

KRELL LOGO (NEW) *
T0-T22 10=96
SECTOR EDIT PATCHED TRACK 2, SECTOR 3
CHANGE ADDRESSES 5B-5C FROM D0 03 TO EA EA

LETTER PERFECT *
T0-T22 10=96, 9=0

LIST HANDLER AND UTILITY *
T1-T11
T0 9=0, A=3, 44=1, 45=D, 50=3
T12-T22.5 STEP .5 D=1, E=F5, F=D7, 10=F7, 45=8, 46=D, 51=1
(See notes for Choplifter)

MAGIC MAILER *
T0-T22

MAGIC WINDOW *
T0-T22

MARS CARS * (DataMost)
T0-T22 10=96

MATH STRATEGY * (Apple)
T0-T22 10=96, 24=96, D=1

MECC (Vol 1 & 2) *
T0-T22
T2 10=96, 9=0

MICROSOFT ADVENTURE *
T0-T22

```
MICRO WAVE *               (Cavalier)
        T0-T22
        T11                3B=1, A=1, 4B=1, 4D=8, 50=1

MILLIKEN SERIES *
        T0-T22

MISSING RING *
        T0-T22             D=1, 24=96, 10=96, 34=1 (DO NOT WRITE PROTECT!)

MIX AND MATCH *
        USE COPY DISK FROM MAIN MENU

MOUSKATTACK *              (Sierra On-Line)
        T0-T22             10=96
        SECTOR EDIT DOS 3.3 PATCHED
          TRACK 18, SECTOR 3
          CHANGE ADDRESS B1 FROM 49 TO 60

MULTI-DISK CATALOG *       (Sensible)
        T0-T8
        T3                 A=1, E=AF, 3B=1, 4B=1, 4D=8, 50=1

MULTI PLAN                 (Microsoft)
        T0-T22             10=96

NIBBLES AWAY I *
        T0-T22

NIBBLES AWAY AUTO LOAD *
        T0-T22             E=DT, 10=97

OLYMPIC DECATHALON *       (Microsoft)
        T0-T22             9=0

OO-TOPOS                   (Sentient)
        T0-T22

OPERATION APOCALYPSE *
        T0-T22             E=DB, F=D5, 10=DE, 8=1

ORBITRON *
        T0-T1              9=0, 31=0
        T1.5-TF.5
        Write Protect Copy!

PEEPING TOM *              (Micro Lab)
        T0
        T1                 E=F5, F=AB, 10=BE, 9=0
        T4-T22

PERSONAL FINANCE MGR. * (Special Delivery)
        T0-T22             10=96

PFS & PFS REPORT           (Software Publishing Corp.)
        USE "COPY DISK" FROM MAIN MENU. AFTER COPYING AND BEFORE USING,
        PUT A TAB OVER THE WRITE PROTECT NOTCH OR YOU WILL TURN
        INTO A LIZARD.
```

```
PHANTOMS FIVE             (Sirius)
       T0                 9=0
       T2-T1C             3A=0, 50=20

PIK (APPLE /// BOOT PROGRAM) *
       USE COPY DISK FROM MAIN MENU

PINBALL CONSTRUCTION *
       T0-T22 (or use Copy Disk)

POOL 1.5 *                (Innovative Design)
       T0-T15

PRISM *
       T0-T22

PRISONER *
       T0-T22

RASTER BLASTER            (Old & New Versions - Budgeco)
       T0                 10=96
       T5-T11  STEP 4  D=1, 9=0, 31=0, A=2, E=AD, F=DE, 3B=40
       T6-T12  STEP 4
       T7.5-TF.5  STEP 4
       T1.5-T3.5  STEP 2

RENDEZVOUS *              (Eduware)
       T0-T23             10=96, 9=0

RENDEZVOUS *
       T0-T22             10=96, 24=96, D=1, 9=0, 31=0

SABOTAGE *
       T0-T22
       T3                 3B=1, A=1, 4B=1, 4D=8, 50=1  (ERROR 6 OK)

SARGON *                  (Hayden)
       T0-T1A             10=F7

SCREENWRITER II *
       Copy Disk from main menu, then Sector Edit
         DOS 3.3 Patched
         Track 3, Sector B
         Change Addresses 94, 95, 96 from 20 00 7F to EA EA EA

SEAFOX *                  (Broderbund)
       SAME AS CHOPLIFTER

SENSIBLE SPELLER *
       T0-T10             10=96, 9=0

SNOGGLE *                 (Broderbund)
       T0-T9              9=0, 8=1

SPACE INVADERS *
       T0-T22             10=96

SPELLING STRATEGY *       (Apple)
       T0-T22             10=96, 24=96, D=1
```

```
SNACK ATTACK  old ver.  (Data Most)
        T0-T12
        SECTOR EDIT DOS 3.2 PATCHED
          TRACK 0 SECTOR 3
          CHANGE ADDRESS 63 FROM 38 TO 18

SNEAKERS *              (Sirius)
        T0              9=0, 10=96, 44=1, 45=10, D=1
        T1.5-TC.5       44=0
        TD.5            44=1

SOFTPORN ADVENTURE *    (On-Line)
        T0-T22          9=0
        T3              3B=1, A=1, 4B=1, 4D=8, 50=1  (ERROR 6 OK)

SPACE EGGS *            (Sirius)
        T0              9=0
        T2-T6
        T11-1A

SPACE INVADERS *
        T0-T22          10=96

SPACE VIKINGS *
        T0-T22

SPECTRE *               (DataMost)
        T0-T2           10=96, 9=0, 8=1
        T3-T22          31=0, E=C5, 10=B5

SPEED READING *
        T0-T22          9=0, 10=96

SPIDER RAID *           (Insoft)
        T0
        T1-T17          A=3, E=92, F=93, 4F=1, 10=95, 44=1, 46=A, 9=0
                        8=1, D=1, 24=96, 3F=1, 34=1, 36=2A, 37=97, 31=0, 43=0
        T1.5-T17.5      E=95, 10=92
        (See Notes for Choplifter)
        (Only Works on New Versions)

SPY´S DEMISE *          (Penquin)
        T1-T11 STEP 2   9=0, 10=96, E=D4
        T0-T12 STEP 2   6=4, 31=0 (ERROR 2 ON TRACK 12 OK)


STARBLASTER *
        T0              10=96, 9=0
        T7-T20 STEP 1.5 E=DF, F=AD, 10=DE

STARBLAZER              (Broderbund)
        SAME AS CHOPLIFTER

STARCROSS *             (Infocom)
        T0-T22          10=96

STELLAR INVADERS *      (Apple)
        T0-T22

STERLING SWIFT PRODUCTS *
        T0-T22
```

STOCK PORTFOLIO SYSTEM *
T3-T22
T0-T2 4=FD, 8=1, 10=AD

UPER TEXT 40/80 *
T0-T22

SWASHBUCKLER * (Data Most)
T0-T22

TAWALA´S LAST REDOUBT * (Broderbund)
T0-T22 D=1

TAX MANAGER * (Microlab)
Use Copy Disk from Main Menu

TAX PREPARER * (Howardsoft)
Use Copy Disk from Main Menu

TEMPLE OF APSHAI *
T0-T22 A=3, 10=96

THRESHOLD (On-Line)
T0-T22
T1-T23 STEP 22 3B=1, A=1, 4B=1, 4D=8, 50=1 (ERROR 6 OK)

TUBE WAY *
T0-T22

TYPING TUTOR * (Microsoft)
USE COPY DISK FROM MAIN MENU

ULTIMA II *
Copy Disk, then Sector Edit
Track 3, Sector 0C
Change Addresses 84, 85, 86 all to EA.

ULTIMA II *
T0-T22 10=96, 9=0, 34=1, 31=0

ULYSSES & GOLDEN FLEECE *
T0-T22 9=0
T3 3B=1, 0A=1, 4B=1, 4D=8, 50=1 ERROR 6 OK

ULYSSES & GOLDEN FLEECE *
COPY DISK FROM MAIN MENU
T3 3B=1, 0A=1, 4B=1, 4D=8, 50=1 ERROR 6 OK

V.C. * (Avlon Hill)
T0-T22

VERSAFORM *
T0-T22

VISICALC (Visicorp)
T0-T16

VISICALC /// (ALL VERSIONS) *
T0-T22 10=96, 24=96, D=1

VISIDEX, VISISCHEDULE, VISITERM, VISITREND/VISIPLOT (Visicorp)
DON´T USE BIT COPY. USE "COPY DISK" FROM MAIN MENU.

VISIFILE (Visicorp)
T0-T22 10=96, 34=1, 36=2A, 37=EB, 3E=2

WARP FACTOR *
T0-T22

WARP FACTOR *
T0
T1-T22 E=DB, F=D5, 10=DE, 8=1

WINDFALL *
T0-T22 10=96

WIZARD & PRINCESS *
T0-T22

WIZARDRY * (Boot Side)
T0-T23 10=96, 24=96, D=1
Note: Write Protect Backup Before Using.

WIZARDRY * (Scenario)
T0-T22 10=96, 24=96, D=1

WIZ MAKER *
T0-T22 D=1, 24=96, 10=96, 34=1, 8=1

ZARGS * (Insoft)
Same as Spider Raid

ZOOM GRAPHICS *
T0-T22 10=96, 9=0

ZOOM GRAPHICS (NEW) *
T0 10=96
T2-T22 STEP 2 9=0, 8=1, 3E=2
T1-T21 STEP 2 E=D4

ZORK I, II, III *
T0-T22 10=96